

# Estudio e instrumentación de una embarcación autónoma

## Trabajo Final de Grado



Facultad de Náutica de Barcelona  
Universidad Politécnica de Cataluña

Trabajo realizado por:  
**Alex Buitrago Cervilla**

Dirigido por:

Rosa Maria Fernández Cantí  
Jordi Fonollosa Magrinyà

Grado en Ingeniería en Sistemas y Tecnología Naval

Barcelona, 27/06/2018

Departamento de Ingeniería de Sistemas, Automática e  
Informática Industrial



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Facultat de Nàutica de Barcelona



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

---

Facultat de Nàutica de Barcelona

Hoja de cortesía



## Agradecimientos

Tengo que agradecer el apoyo incondicional de mi familia. A pesar de haberles causado un gran dolor de cabeza, tanto por asuntos personales como con el proyecto, han estado a mi lado en todo momento, dando opiniones apoyo y consejos que me han servido de gran ayuda para realizar con éxito el trabajo. En especial a mi padre y a mi madre, por ser los que más me han ayudado, invirtiendo su tiempo y sus recursos.

Me gustaría agradecer a todas y cada una de las personas, familiares y amigos, que me han hecho valorar el gran trabajo que estaba haciendo, por darme todo el apoyo posible y ofrecer cualquier ayuda que pudiesen. Especialmente a mi tía Silvia, mi primo Jordi, mis amigos Alex García, Jhonny Caballero, Christian Velasco, mi hermana Sandra, al profesor Sergio Romero y al Club Náutico de Badalona.

Agradecer a los profesores y tutores implicados en el proyecto, y a los profesores que me enseñaron durante el grado todo el conocimiento necesario para lograr un sueño.

Tengo que agradecer a todas y cada una de las personas, que directa o indirectamente han intervenido en este trabajo, ofreciendo pequeñas ideas, que a priori parecían solamente posibilidades, pero eran un rayo de esperanza para el proyecto.

Necesitaría muchas páginas para agradecer a todas y cada una de las personas que me han ayudado. Gracias de corazón.



## Resumen

El siguiente proyecto desarrolla la creación de un prototipo de embarcación autónoma y la explicación de la evolución de los buques autónomos desde su origen.

Empezando por el diseño del casco utilizando programas avanzados como Rhinoceros y la comprobación de la efectividad de la estructura en el agua mediante Maxsurf.

Una vez realizados los planos 3D de las palas, se imprimen en impresoras 3D los distintos módulos para posteriormente unirlos a través de elementos que aportan dureza y resistencia.

Teniendo la estructura externa del buque, se procede a la elección de los dispositivos que se encargaran de la navegación totalmente autónoma, la creación del circuito encargado de sincronizar todos y cada uno de los elementos, y la programación del código, todo se llevará a cabo gracias a la plataforma Arduino.

Tras realizar físicamente el buque, se elaboran pruebas de comportamiento de la embarcación y esquive de obstáculos con la intención de conocer las características del buque. El resultado final ha sido un éxito a pesar de todos los contratiempos que han ido sucediendo durante las pruebas, se ha logrado evitar la colisión de la nave en dos casos, manteniéndose dentro de los límites de una piscina y, ir y volver a un punto sin ningún incidente.

Finalmente, se explica la situación actual de los buques autónomos y su posible evolución en los próximos años.

## Abstract

The following project develops the creation of a prototype of an autonomous ship and the explanation of the evolution of autonomous vessels from their origin.

Starting with helmet design using advanced programs such as Rhinoceros and checking the effectiveness of the structure on the water using Maxsurf.

Once the 3D plans of the blades have been made, the different modules are printed on 3D printers and then, are joined together using elements that provide hardness and resistance.

Having the external structure of the ship, we proceed to the election of the devices that will be taking the responsibility for the totally autonomous navigation, the creation of the circuit in charge of synchronizing each and every one of the elements, and the programming of the code, everything will be carried out thanks to the Arduino platform.

After the build up of the vessel, tests of the behaviour of the vessel and avoidance of obstacles are made with the intention of knowing the characteristics of the vessel. The final result has been a success despite all the setbacks that have been happening during the tests, it has been possible to avoid the collision of the ship in two cases, staying within the limits of a pool and, go and return to a point without no incident

Finally, the current situation of autonomous vessels and their possible evolution in the coming years is explained.





## **Prefacio**

El trabajo final de grado, es un proyecto muy importante en el que demuestras el conocimiento adquirido a lo largo del grado universitario.

Al tener tanta libertad para escoger un tema, quería realizar una propuesta que implicase un reto. Barajando todas las posibilidades que estaban expuestas en la página de propuestas de TFG de la Facultad náutica de Barcelona, me causaron curiosidad diversos proyectos que estaban relacionados con la profesora Rosa M. Fernández, así que me puse en contacto con ella.

Los puntos principales para elegir el TFG fueron, ampliar conocimientos desde un aspecto práctico, ya que durante la trayectoria universitaria la mayoría de los trabajos han sido teóricos, y realizar proyectos con la placa Arduino, que durante el grado no hemos elaborado experimentos con este tipo de hardware.

Teniendo en cuenta los puntos anteriormente nombrados, la idea de crear un buque autónomo resultaba un gran reto como trabajo de final de grado. De esta manera, la profesora Rosa M. Fernández estuvo de acuerdo con el desarrollo de esta propuesta, y pasó a ser tutora del proyecto.

Además, para una mayor profundización para el desarrollo de la electrónica y el código, me puse en contacto con el profesor Jordi Fonollosa, al explicarle en qué consistía el trabajo se interesó en el proyecto y formó parte de éste como co-tutor.

# Tabla de contenidos

AGRADECIMIENTOS	II
RESUMEN	IV
ABSTRACT	V
PREFACIO	VII
TABLA DE CONTENIDOS	VIII
LISTA DE FIGURAS	XI
LISTA DE TABLAS	XV
<b>CAPÍTULO 1. INTRODUCCIÓN</b>	<b>1</b>
1.1 MOTIVACIÓN DEL TRABAJO	1
1.2 OBJETIVOS DEL TRABAJO	2
1.3 ORGANIZACIÓN DE LA MEMORIA	3
<b>CAPÍTULO 2: PRESENTE Y FUTURO DE LAS EMBARCACIONES AUTÓNOMAS</b>	<b>4</b>
2.1 CASOS DE EMPRESAS ACTUALES	4
2.1.1 YARA	4
2.1.2 ROLLS ROYCE	7
2.2 ACTUACIÓN DE LAS ORGANIZACIONES INTERNACIONALES	8
2.2.1 COMITÉ DE SEGURIDAD MARÍTIMA (MSC)	8
2.2.2 MARITIME UK	9
<b>CAPÍTULO 3: DISEÑO DE LA EMBARCACIÓN</b>	<b>10</b>
3.1 INTRODUCCIÓN	10
3.2 RHINOCEROS	12
3.3 MAXSURF	14
3.4 IMPRESIÓN	15
<b>CAPÍTULO 4: ENSAMBLAJE DEL CASCO</b>	<b>18</b>
4.1 UNIÓN DE LAS PIEZAS DEL CASCO	18
4.1.1 PEGAMENTO BICOMPONENTE	18
4.1.2 MASILLA DE POLIÉSTER	19
4.1.3 IMPRIMACIÓN	21
4.1.4 PINTURA	22

<b>4.2 UNIÓN DE LAS DOS PALAS</b>	<b>24</b>
<b>4.3 CAJA</b>	<b>25</b>
<b>4.4 ARBOTANTES</b>	<b>27</b>
<b><u>CAPÍTULO 5: INSTRUMENTACIÓN Y CONTROL</u></b>	<b><u>28</u></b>
<b>5.1 ELEMENTOS PARA OBTENER INFORMACIÓN DEL MEDIO</b>	<b>28</b>
5.1.1 SENSORES INFRARROJOS	28
5.1.2 SENSORES DE ULTRASONIDOS	29
<b>5.2 ELEMENTOS PARA ADMINISTRAR LA INFORMACIÓN</b>	<b>30</b>
5.2.1 ARDUINO	30
5.2.2 RASPBERRY PI	30
<b>5.3 CÓDIGO</b>	<b>31</b>
<b>5.4 ELEMENTOS QUE ACTÚAN SEGÚN EL CÓDIGO</b>	<b>31</b>
<b><u>CAPÍTULO 6: PLAN DE PRUEBAS Y RESULTADOS</u></b>	<b><u>34</u></b>
<b>6.1 PRUEBA DE FLOTABILIDAD</b>	<b>34</b>
<b>6.2 PRUEBA DE ULTRASONIDOS</b>	<b>35</b>
<b>6.3 PRUEBA DE FRENO</b>	<b>38</b>
<b>6.4 PRUEBA DE AVANCE</b>	<b>39</b>
<b>6.5 PRUEBA DE GIRO</b>	<b>40</b>
<b>6.6 MODIFICACIONES</b>	<b>42</b>
<b>6.7 PRUEBA DE EVITAR LA COLISIÓN CON EL BORDE DE LA PISCINA</b>	<b>43</b>
<b>6.8 PRUEBA DE LLEGAR A UN LUGAR Y VOLVER</b>	<b>48</b>
<b>6.9 PRUEBA DE EVITAR LA COLISIÓN CON OBSTÁCULOS</b>	<b>49</b>
<b>6.10 DIFICULTADES</b>	<b>53</b>
<b>6.11 ALTERNATIVAS</b>	<b>54</b>
DURANTE EL DESARROLLO DE CIRCUITOS Y CÓDIGOS HUBIERON ALGUNOS QUE NO SE LLEVARON A CABO PORQUE NO LLEGARON A FUNCIONAR. COMO POR EJEMPLO, EL CIRCUITO DE CONTROL VARIABLE DEL MOTOR DC A TRAVÉS DE UN TRANSISTOR	54
<b><u>CAPÍTULO 7: PRESUPUESTO</u></b>	<b><u>56</u></b>
<b>7.1 MÓDULOS</b>	<b>56</b>
<b>7.2 ELEMENTOS DE EXTERNOS</b>	<b>57</b>
7.2.1 PREPARACIÓN DEL CASCO A TRAVÉS DE PEGAMENTOS Y SUSTANCIAS	57
7.2.2 ELEMENTOS DE SUJECIÓN	58
<b>7.3 MATERIAL ELECTRÓNICO Y FUNCIONAL</b>	<b>58</b>

<b>CAPÍTULO 8: CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN</b>	<b>60</b>
<b>BIBLIOGRAFÍA Y REFERENCIAS</b>	<b>62</b>
<b>ANEXOS</b>	<b>65</b>
<b>A.1 ALARMA MEDIANTE ULTRASONIDOS A CIERTA DISTANCIA</b>	<b>65</b>
<b>A.2 MODIFICACIÓN DE LA VELOCIDAD DE LOS MOTORES CON PWM Y DOBLE PUENTE H</b>	<b>67</b>
<b>A.3 GIRO CON LOS DOS MOTORES</b>	<b>73</b>
<b>A.4 FRENO CON LOS DOS MOTORES INVERTIDOS</b>	<b>76</b>
<b>A.5 CÓDIGO PARA EVITAR LA COLISIÓN CON EL BORDE DE LA PISCINA</b>	<b>80</b>
<b>A.6 CÓDIGO PARA EVITAR COLISIONES Y SEGUIR LA RUTA</b>	<b>85</b>

# Lista de figuras

<b>FIGURA 1. RUTA REALIZADA POR LOS CAMIONES DESDE HEROYA HASTA BREVIK.....</b>	<b>5</b>
<b>FIGURA 2. OPERACIONES DE CARGA Y DESCARGA .....</b>	<b>6</b>
<b>FIGURA 3. FIORDOS DE TRONHEIM .....</b>	<b>6</b>
<b>FIGURA 4. EVOLUCIÓN DE LOS BUQUES SEGÚN ROLLS-ROYCE.....</b>	<b>7</b>
<b>FIGURA 5. GRÁFICA MANGA-ESLORA .....</b>	<b>11</b>
<b>FIGURA 6. GRÁFICA CALADO-MANGA.....</b>	<b>11</b>
<b>FIGURA 7. GRÁFICA CALADO-ESLORA .....</b>	<b>11</b>
<b>FIGURA 8. PALA DEL CASCO EN RHINOCEROS .....</b>	<b>13</b>
<b>FIGURA 9. CASCO SECCIONADO PARA LA IMPRESIÓN .....</b>	<b>13</b>
<b>FIGURA 10. PRUEBA DE FLOTABILIDAD EN MAXSURF.....</b>	<b>14</b>
<b>FIGURA 11. MÓDULOS POR SEPARADO .....</b>	<b>17</b>
<b>FIGURA 12. PALAS UNIDAS POR PEGAMENTO BICOMPONENTE .....</b>	<b>19</b>
<b>FIGURA 13. PEGAMENTO BICOMPONENTE .....</b>	<b>19</b>
<b>FIGURA 14. MASILLA DE POLIÉSTER SIN LIJAR .....</b>	<b>20</b>
<b>FIGURA 15. CASCO CON LA MASILLA DE POLIÉSTER LIJADA .....</b>	<b>21</b>
<b>FIGURA 16. MASILLA FINA DE POLIÉSTER.....</b>	<b>21</b>
<b>FIGURA 17. IMPRIMACIÓN MULTIUSOS.....</b>	<b>22</b>
<b>FIGURA 18. CASCO DEL BUQUE CON LA IMPRIMACIÓN .....</b>	<b>22</b>

<b>FIGURA 19. PINTURA TITAN YATE.....</b>	<b>23</b>
<b>FIGURA 20. CASCO PINTADO .....</b>	<b>23</b>
<b>FIGURA 21. UNIÓN DE LAS PALAS CON LAS PIEZAS DE MADERA .....</b>	<b>25</b>
<b>FIGURA 22. MEDIDAS DE LA PLACA DE PVC.....</b>	<b>26</b>
<b>FIGURA 23. CAJA.....</b>	<b>26</b>
<b>FIGURA 24. ARBOTANTE .....</b>	<b>27</b>
<b>FIGURA 25. SENSOR DE ULTRASONIDOS HC-SR04 .....</b>	<b>29</b>
<b>FIGURA 26. PLACA ARDUINO UNO .....</b>	<b>31</b>
<b>FIGURA 27. MOTOR DE 12V .....</b>	<b>32</b>
<b>FIGURA 28. DISPOSITIVO DOBLE PUENTE H L298N .....</b>	<b>33</b>
<b>FIGURA 29. PRUEBA DE FLOTABILIDAD CON CASCO .....</b>	<b>34</b>
<b>FIGURA 30. PRUEBA DE FLOTABILIDAD CON CASCO Y CAJA.....</b>	<b>35</b>
<b>FIGURA 31. CÓDIGO DE ULTRASONIDOS.....</b>	<b>36</b>
<b>FIGURA 32. VALORES MÁXIMOS DE LA PRUEBA DE ULTRASONIDOS.....</b>	<b>37</b>
<b>FIGURA 33. VALORES MÍNIMOS DE LA PRUEBA DE ULTRASONIDOS.....</b>	<b>37</b>
<b>FIGURA 34. BUCLE LOOP .....</b>	<b>39</b>
<b>FIGURA 35. REPRESENTACIÓN DEL BUQUE EN LA PISCINA DURANTE LA PRUEBA DE AVANCE .....</b>	<b>40</b>
<b>FIGURA 36. COMPORTAMIENTO DEL BUQUE CON LA CORRIENTE .....</b>	<b>41</b>
<b>FIGURA 37. BUCLE EN LA PRUEBA DE GIRO .....</b>	<b>41</b>

<b>FIGURA 38. SENSORES DE ULTRASONIDOS INSTALADOS .....</b>	<b>42</b>
<b>FIGURA 39. DIMENSIONES DE LA PISCINA.....</b>	<b>43</b>
<b>FIGURA 40. PISCINA NÚMERO 1 .....</b>	<b>44</b>
<b>FIGURA 41. PISCINA NÚMERO 2 .....</b>	<b>44</b>
<b>FIGURA 42. PISCINA NÚMERO 3 .....</b>	<b>45</b>
<b>FIGURA 43. DIAGRAMA DE ESTADOS INICIAL .....</b>	<b>46</b>
<b>FIGURA 44. COMPORTAMIENTO DE LA EMBARCACIÓN DURANTE LA PRUEBA .....</b>	<b>47</b>
<b>FIGURA 45. VARIABLES DEFINIDAS .....</b>	<b>49</b>
<b>FIGURA 46. DIAGRAMA DE ESTADOS PARA EVITAR OBSTÁCULOS .....</b>	<b>50</b>
<b>FIGURA 47. ACCIONES DEL BUQUE DURANTE EL ESQUIVE DE OBSTÁCULOS .....</b>	<b>52</b>
<b>FIGURA 48. CIRCUITO DE CONTROL VARIABLE DEL MOTOR DC.....</b>	<b>55</b>
<b>FIGURA 49. ALARMA MEDIANTE ULTRASONIDOS A CIERTA DISTANCIA .....</b>	<b>65</b>
<b>FIGURA 50. CÓDIGO DE ULTRASONIDOS.....</b>	<b>66</b>
<b>FIGURA 51. CONEXIÓN PARA MOTORES MEDIANTE PWM .....</b>	<b>67</b>
<b>FIGURA 52. CÓDIGO PARA MOTORES MEDIANTE PWM .....</b>	<b>71</b>
<b>FIGURA 53. CÓDIGO DE GIRO CON LOS DOS MOTORES .....</b>	<b>75</b>
<b>FIGURA 54. CÓDIGO DE FRENO CON LOS DOS MOTORES .....</b>	<b>78</b>
<b>FIGURA 55. CÓDIGO PARA EVITAR LA COLISIÓN CON EL BORDE .....</b>	<b>84</b>
<b>FIGURA 56. CÓDIGO PARA EVITAR COLISIONES Y SEGUIR LA RUTA.....</b>	<b>91</b>





# Lista de tablas

<b>TABLA 1. VALORES DE ESLORA, MANGA Y CALADO PARA DIVERSOS MODELOS DE CATAMARES</b>	<b>10</b>
<b>TABLA 2. DIMENSIONES DE LA MAQUETA</b>	<b>12</b>
<b>TABLA 3. VALORES DE FLOTABILIDAD DEL CASCO</b>	<b>15</b>
<b>TABLA 4. DATOS DE IMPRESIÓN</b>	<b>16</b>
<b>TABLA 5. CANTIDAD DE TIEMPO Y MATERIAL</b>	<b>16</b>
<b>TABLA 6. DIMENSIONES DE LOS SOPORTES</b>	<b>24</b>
<b>TABLA 7. DIMENSIONES DE LA CAJA</b>	<b>25</b>
<b>TABLA 8. COMPARACIÓN TEÓRICO-PRÁCTICA DE LOS SENSORES DE ULTRASONIDOS</b>	<b>37</b>
<b>TABLA 9. PRECIO TOTAL DE LA IMPRESIÓN DE LOS MÓDULOS</b>	<b>56</b>
<b>TABLA 10. MATERIAL DE UNIÓN DEL CASCO</b>	<b>58</b>
<b>TABLA 11. MATERIAL DE SUJECCIÓN</b>	<b>58</b>
<b>TABLA 12. MATERIAL ELECTRÓNICO</b>	<b>59</b>



# Capítulo 1. Introducción

## 1.1 Motivación del trabajo

En el ámbito de la ingeniería naval uno de los grandes retos del siglo XXI es la construcción de buques que se puedan manejar de forma autónoma. Los impulsores de esta tendencia son las grandes empresas navieras, para las cuales la reducción de costes es un factor primordial a tener en cuenta. Según [5] se estima que actualmente un 44% del dinero invertido en buques se debe a la tripulación, tanto por su salario, como por su acomodación. Además, el 96% de los accidentes provienen de errores humanos.

Esta idea tiene diversas consecuencias que incluyen no solo la modificación de los diseños del buque, sino también la eliminación de los puestos de trabajo de miles de personas y cambios en la estructura de los puertos, en los que habrá que crear torres de control que serán las responsables de las acciones remotas del buque. Las predicciones más optimistas afirman que en los próximos años, el sector evolucionará de manera que el trabajo de marinero no existirá, y toda la tripulación pasará a trabajar en torres de control y puertos.

Los trabajos más relevantes en este campo provienen de las empresas Rolls Royce o Yara [13], que pretenden implementar, a partir del año 2035, sistemas y sensores que permitan el control del buque en su totalidad. Por su lado, una de las sociedades de clasificación más importantes, Lloyd's Register, ha emitido directrices para operaciones sin tripulación, por lo que prevé un futuro con este tipo de buques [5].

En definitiva, nos encontramos ante un escenario nuevo, donde el reto es la implementación de sistemas y sensores que permitan el control del buque en su totalidad y el análisis de su viabilidad [5].

<sup>1</sup> OMI: La Organización Marítima Internacional es un organismo especializado de las Naciones Unidas que promueve la cooperación entre Estados y la industria de transporte para mejorar la seguridad marítima y para prevenir la contaminación marina.

## 1.2 Objetivos del trabajo

El objetivo global del presente trabajo fin de grado es observar y analizar cuáles son los límites que ofrecen este tipo de buques. En una primera aproximación este análisis se realizará a partir de la información que ofrecen las empresas y estudios disponibles pero, sobre todo, se llevará a cabo de manera práctica mediante la implementación de un prototipo de catamarán operado de manera autónoma.

Así pues, el objetivo principal es la construcción de un modelo a escala de un buque autónomo que permita estudiar el comportamiento en situaciones de esquivar obstáculos.

Por su parte, los objetivos específicos son los siguientes:

### Estado del arte

- Conocer el estado actual de los buques autónomos y su evolución.

### Construcción

- Diseñar de la forma más óptima posible el casco del buque, realizando cálculos de flotabilidad previos.
- Elegir los productos más adecuados para la unión de las piezas.
- Ejecutar con éxito el ensamblaje del casco mediante los elementos de adherencia.

### Electrónica

- Elegir los dispositivos más adecuados que servirán como instrumentación y control de la nave. Y posteriormente, diseñar e implementar el circuito de condicionamiento de los sensores para detectar objetos a distancia.
- Diseñar e implementar correctamente el código encargado de evitar la colisión del buque con cualquier objeto que se sitúe en su trayectoria mediante una plataforma Arduino.

### Pruebas de comportamiento

- Efectuar las pruebas necesarias con la intención de conocer el comportamiento del barco y de los elementos de navegación.
- Lograr que el buque sea capaz de evitar la colisión en todo momento con cualquier objeto que se encuentre a su alrededor.

### **1.3 Organización de la memoria**

La presente memoria se organiza de la siguiente manera:

En el Capítulo 2. Presente y futuro de las embarcaciones autónomas, se describen los trabajos actuales de las empresas Yara y Rolls Royce y se presenta brevemente el marco normativo establecido por la Organización Internacional Marítima (OMI) en el ámbito de las embarcaciones autónomas.

El Capítulo 3. Diseño de la embarcación, aborda el proceso de diseño del casco a través de programas especializados, y presenta los cálculos previos a la impresión del casco.

En el Capítulo 4. Ensamblaje del casco, se explican los elementos encargados de unir cada uno de los módulos y aportar la resistencia estructural al casco.

El Capítulo 5. Instrumentación y control, trata de la elección de los componentes para el circuito Arduino según su función.

En el capítulo 6. Plan de pruebas y resultados, se estudia el comportamiento del buque en unas pruebas específicas con la intención de lograr el objetivo de no colisionar con ningún obstáculo.

El Capítulo 7. Presupuesto, muestra todos los gastos en cada una de las etapas del proyecto.

Finalmente, en el Capítulo 8, se realiza un balance del trabajo y se presentan las conclusiones y líneas futuras de investigación.

En los anexos se dan más detalles de los circuitos y se listan los códigos realizados para las distintas funcionalidades de la maqueta.

## Capítulo 2: Presente y futuro de las embarcaciones autónomas

### 2.1 Casos de empresas actuales

Los buques autónomos a día de hoy son prácticamente inexistentes, empresas como Rolls Royce o Yara han tomado la iniciativa, y pretenden implementar sistemas y sensores que permitan el control del buque en su totalidad en los próximos años.

#### 2.1.1 Yara

La compañía noruega Yara [13] se ha unido con la empresa de tecnología marítima Kongsberg para diseñar y construir el primer buque de contenedores eléctrico y autónomo del mundo. Transportará productos químicos y fertilizantes desde la planta de producción de Yara en Heroya hasta las ciudades de Brevik y Larvik, todas en Noruega.

La previsión esperada por Yara comienza con la primera prueba en alta mar que se realizará a finales de 2018 y estará tripulada. Más adelante, la compañía espera hacer una evolución a la operación remota en 2019 y finalmente, ceder totalmente el control al propio buque en 2020.

A pesar de la pérdida de puestos de trabajo, hay un beneficio para el medio ambiente: hasta ahora, el transporte de productos de Yara requería 40.000 viajes de camiones al año. En cambio, la misma ruta bordeando la costa con el barco estará libre de emisiones de NO<sub>x</sub> y CO<sub>2</sub>.



Figura 1. Ruta realizada por los camiones desde Herøya hasta Brevik [13]

La construcción del buque por parte de Yara no resultará barata, el proyecto se estima en 25 millones de dólares, una cifra tres veces superior a la habitual en la construcción de cargueros. La ventaja se encuentra en el ahorro de costes a largo plazo.

Según sus cálculos, un barco de este tipo, con capacidad para 100 contenedores, que no necesita tripulación ni necesidad de llenar depósitos de combustibles, permitiría ahorrar hasta un 90% de los costes operacionales anuales.

Como es un buque que realiza un trayecto de 10 kilómetros, no requiere de una alta autonomía para realizar con éxito la ruta. El buque obtendría la energía necesaria del mismo puerto de Herøya o Brevik, dependiendo de la necesidad.

La misma compañía muestra como efectuarían las operaciones de carga y descarga a través de un vídeo en su página web [15]:





Figura 2. Operaciones de carga y descarga. [13]

La zona de pruebas para buques autónomos, no será un inconveniente, debido a que la Autoridad Marítima de Noruega y la Administración de Costas de Noruega han firmado un acuerdo para que el fiordo de Trondheim se convierta en la primera zona de pruebas del mundo específicamente destinada a los buques autónomos. El escaso tráfico marítimo que registra esta zona del país, sumado a sus particularidades geográficas y la presencia de diversas industrias a lo largo de la línea de costa, así como la apuesta gubernamental por las nuevas tecnologías, lo convierten en un sitio idóneo para comprobar la viabilidad de algunos de los navíos más sofisticados del planeta.

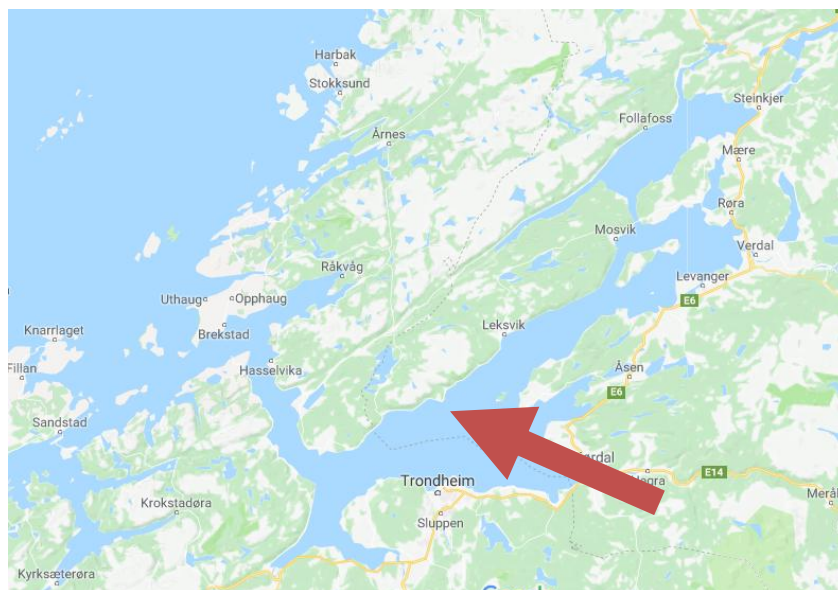


Figura 3. Fiordos de Trondheim [13]

### 2.1.2 Rolls Royce

Por otra parte, Rolls Royce está trabajando en soluciones que permitan alcanzar la automatización absoluta en los buques. De hecho, la compañía británica ya ha avanzado sus planes para desarrollar un buque de estas características con una autonomía de 3500 millas náuticas, 60 metros de eslora, velocidades superiores a los 25 nudos y la posibilidad de permanecer más de tres meses en alta mar.

El proyecto Rolls Royce propone la supresión de la tripulación en los buques, que pasarían a formar parte del control remoto de estos desde centros de control situados en Finlandia. El modelo mostrado por la empresa británica, de momento es futurista, aunque asegura que dentro de 10 a 15 años se puede cumplir.

Como consecuencia de estos hechos, el barco sería un 5% más ligero y quemaría un 15% menos de combustible. Además, los costes que suponen los marineros no se limitan a los salarios, también a los dormitorios de la tripulación, las unidades de aire acondicionado, el puente de mando... Todo esto representa el 44% de los costes del barco.

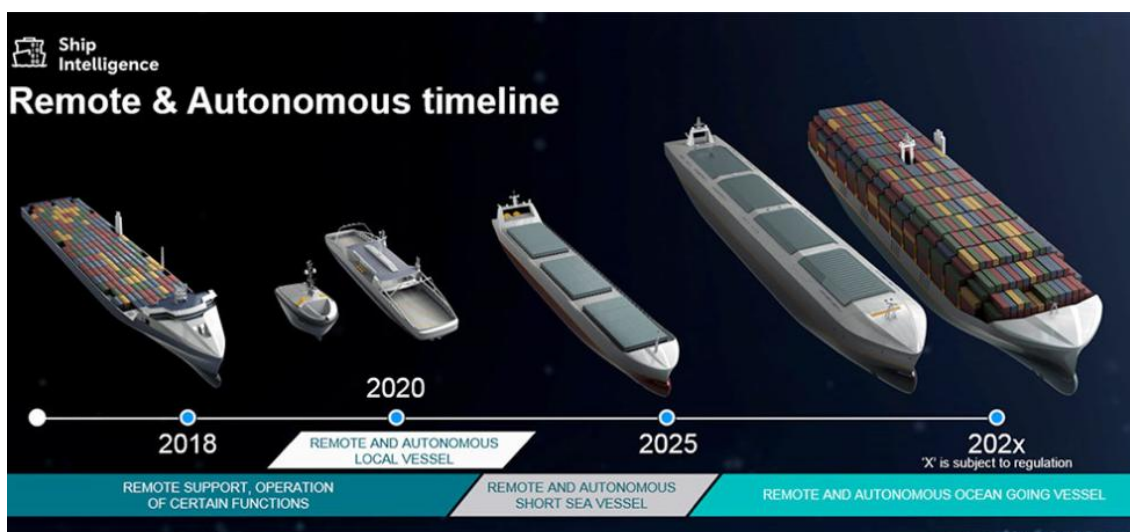


Figura 4. Evolución de los buques según Rolls-Royce [25]

Además, Rolls-Royce ha firmado un acuerdo con Google para desarrollar sus sistemas inteligentes de conocimiento y hacer más seguras sus naves, consiguiendo hacer realidad la navegación autónoma. El acuerdo, permite a Rolls-Royce el uso del *Cloud Machine Learning Engine* de Google [17] para mejorar el entrenamiento del sistema de clasificación de objetos basado en inteligencia artificial (AI) de la compañía, diseñado para detectar, identificar y realizar el seguimiento de los objetos que puede encontrar un barco cuando navega por el mar.

Por otra parte, es muy probable que en una primera fase no se vean grandes buques mercantes completamente automatizados, sino que el cambio se vaya introduciendo al igual que la compañía noruega Yara, paulatinamente en pequeños trayectos y de forma escalonada.

Por ejemplo, en un ferry con ruta fija los sistemas de navegación serían autónomos y se mantendría la tripulación de servicio para atender a los pasajeros. Además, los sistemas de control irán combinando la navegación remota desde tierra con soluciones de inteligencia artificial que permitan a las embarcaciones reaccionar ante situaciones imprevistas tales como tormentas u obstáculos en su camino.

Debido a la eliminación de la tripulación, los buques autónomos serán naves muy distintas a las actuales, no solo en la funcionalidad, sino en su diseño. Estructuras destinadas hasta ahora a la marinería como el puente de mando desaparecerían, mejorando su aerodinámica. La desaparición de las cabinas de la tripulación también afectaría en la optimización de las bodegas de carga.

A pesar de que los sindicatos que duden de que la tecnología pueda reemplazar a marineros experimentados y que la Organización Marítima Internacional (OMI) prohíbe las operaciones sin tripulación [18], hay que tener en cuenta que los sensores necesarios para la creación de buques autónomos ya están disponibles en el mercado. Además, una de las sociedades de clasificación más importantes, Lloyd's Register ya ha emitido directrices para operaciones sin tripulación.

## 2.2 Actuación de las organizaciones internacionales

### 2.2.1 Comité de Seguridad Marítima (MSC)

La Organización Marítima Internacional (OMI), el órgano regulador del transporte marítimo internacional, ha iniciado la tarea para determinar el modo de abordar en sus instrumentos la navegación de los buques marítimos autónomos de superficie (MASS<sup>2</sup>, *maritime autonomous surface ships*) en condiciones de seguridad, protección y ambientalmente racionales. [19]

El órgano técnico de más antigüedad de la OMI, el Comité de seguridad marítima (MSC), ha refrendado un marco de trabajo para el estudio exploratorio sobre la reglamentación, como labor en curso, que incluye la definición preliminar de los MASS y los grados de autonomía así como la metodología para el estudio y el plan de trabajo.

A los efectos del estudio exploratorio sobre la reglamentación, se entiende por “buque marítimo autónomo de superficie (MASS)” todo buque que, en diversos grados, puede navegar sin depender de la interacción humana.

Para facilitar el proceso del estudio exploratorio sobre la reglamentación, los grados de autonomía se organizan de la siguiente forma:

- -Buque con procesos automatizados y apoyo en la toma de decisiones. La gente de mar está a bordo para operar y controlar los sistemas y las funciones de a bordo. Algunas operaciones pueden estar automatizadas.
- -Buque controlado a distancia con gente de mar a bordo: El buque se controla y opera desde otro emplazamiento, pero hay gente de mar a bordo.
- -Buque controlado a distancia sin gente de mar a bordo: El buque se controla y opera desde otro emplazamiento. No hay gente de mar a bordo.
- -Buque totalmente autónomo: El sistema operativo del buque es capaz de tomar decisiones y de determinar acciones por sí mismo.

Los MASS podrían navegar con varios grados de autonomía durante un mismo viaje.

El MSC, que se reunió en su 99º periodo de sesiones (16-25 de mayo del 2018) [19], constituyó un grupo de correspondencia sobre MASS para probar el marco de trabajo del estudio exploratorio acordado en el periodo de sesiones y, especialmente, la metodología, e informar de nuevo al Comité durante su próximo período de sesiones del 3 al 7 de diciembre de 2018.

### 2.2.2 Maritime UK

Por otra parte, Maritime UK ha publicado un nuevo Código para el diseño, construcción y operación de sistemas marítimos autónomos durante el tercer congreso celebrado por *MAS Regulatory Working Group*, en Southampton.

En la búsqueda de proporcionar una guía práctica para el diseño, construcción y operación segura para barcos autónomos y semi-autónomos de eslora inferior a 24 m, este código ofrecerá directrices mientras se desarrolla un marco regulatorio más detallado para los sistemas autónomos.

El Código se actualizará cuando la guía del análisis de la normativa de la Organización Marítima Internacional (OMI) sea publicada y el comité de seguridad marítima (MSC) desarrolle políticas para satisfacer las necesidades de las demandas tecnológicas, comerciales y regulatorias venideras.

La evolución de esta tecnología tendría una transición desde buques autónomos con tripulación para controlar los procesos, posteriormente al año, que pueda controlarse remotamente, y finalizar, en una navegación del 100% de autonomía.

<sup>2</sup> MASS: Se entiende Maritime Autonomous Surface Ships o Buque Autónomo de Superficie a todo buque que, en diversos grados, puede navegar sin depender de la interacción humana.

## Capítulo 3: Diseño de la embarcación

### 3.1 Introducción

Antes de realizar la maqueta del buque, se debe conocer qué tipo de embarcación es la adecuada para realizar el proyecto y sus dimensiones. Se ha escogido el catamarán debido a que este buque obtendría buenos resultados de estabilidad, ya que al tener dos palas como casco, facilitaría y simplificaría los cálculos de estabilidad como los que se tendrían que haber elaborado con otro tipo de embarcación.

Para dimensionar la maqueta, lo más efectivo resultaba recolectar datos de catamaranes a escala real que se hallasen en internet, con esta información se puede escalar los valores según nuestra necesidad.

La información encontrada ha sido la siguiente [19]:

Dimensiones Catamaranes (m)			
Nombre de los catamaranes	Eslora (L)	Manga (B)	Calado ( C )
Lagoon 42	12,8	7,7	1,25
Alliura Privilege 495	14,95	7,33	1,35
Fountaine MALDIVES 32	9,6	5,3	0,9
Lagoon 500	15,5	8,5	1,4
Fountaine Victoria 67	20,32	9,5	1,55
Lagoon 52	15,85	8,6	1,5
Catamaran Trans cat 42	11,95	5,4	1,3
Lagoon 380	11,55	6,53	1,15
Lagoon 560	17,07	9,44	1,5
Alliura Privilege 495	14,95	7,33	1,35
Passion 32	9,95	5,95	0,95
Lagoon 440	13,61	7,7	1,3
Lagoon 52 F	15,6	8,74	1,55
Lagoon 39	11,63	6,83	1,2
Fountaine Highland 35 Pilot	10,35	4,45	0,95
Sunreef Yachts 62	18,9	9,2	1,58
Bali Catamarans 4.3 Loft	13,10	7,12	1,20

Tabla 1. Valores de eslora, manga y calado para diversos modelos de catamaranes

Posteriormente, se realizaron gráficos relacionando los valores de manga-eslora; calado-manga; calado-eslora, y se agregaron las líneas de tendencia correspondientes a cada uno mediante la función Agregar línea de tendencia.

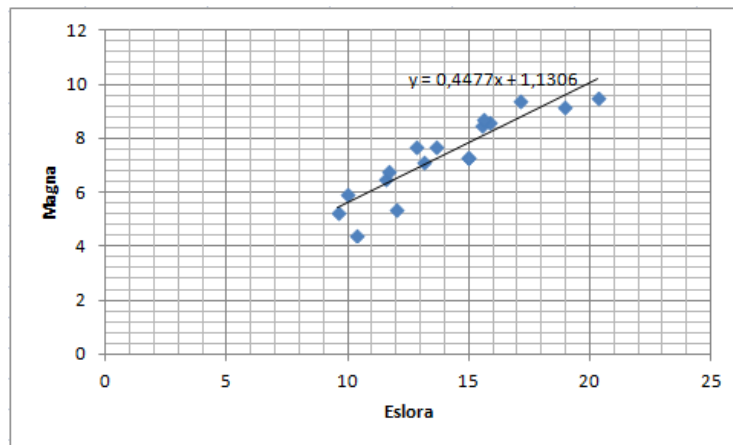


Figura 5. Gráfica Manga-Eslora

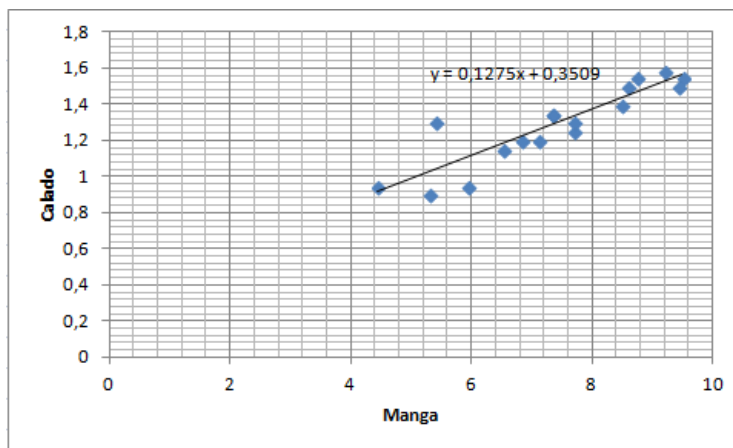


Figura 6. Gráfica Calado-Manga

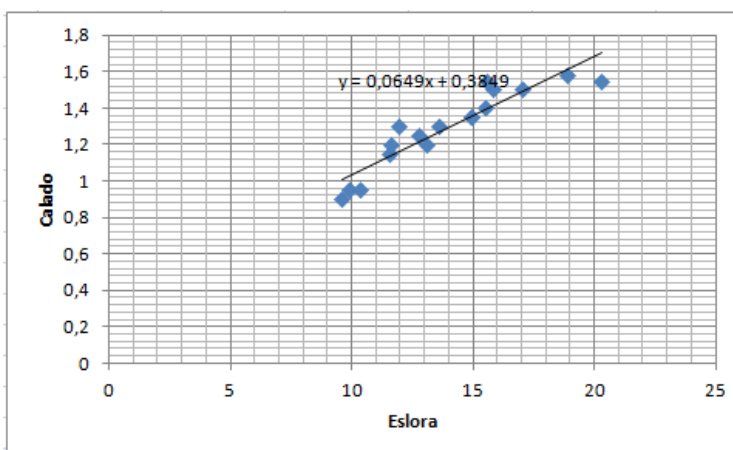


Figura 7. Gráfica Calado-Eslora

Una vez se obtienen las rectas de regresión, se aplican las fórmulas para una embarcación de 10 metros y se escalan a 1/12.

Las rectas de regresión obtenidas son las siguientes:

$$B = 0,4477 \times L + 1,1306 \quad (1)$$

$$C = 0,1275 \times B + 0,3509 \quad (2)$$

$$C = 0,0649 \times L + 0,3849 \quad (3)$$

donde  $B$  es la manga,  $C$  es el calado y  $L$  es la eslora, todas ellas expresadas en metros.

Escalar:

Los valores resultantes para una embarcación de 10 metros se deben escalar a una proporción adecuada a la cantidad de recursos que hay disponibles. La escala elegida ha sido de 1/12, ya que un buque de esas dimensiones tenderá a comportarse como uno con proporciones reales. Si hubiese sido más pequeño, factores como el viento y la corriente hubiesen afectado negativamente al catamarán durante las pruebas.

Escala	0,08
L=	0,83 m
B=	0,47 m
C=	0,09 m

Tabla 2. Dimensiones de la maqueta

### 3.2 Rhinoceros

Una vez se han escalado los valores obtenidos de las rectas de regresión, se traspasan al programa Rhinoceros con la intención de dimensionar el catamarán. Este *software* permite diseñar con más flexibilidad y precisión, en comparación con otros programas como MaxSurf, ya que Rhinoceros es un programa especializado en el diseño. En cambio, MaxSurf está enfocado en el comportamiento de los buques en la mar y no tiene tanta facilidad ni



versatilidad a la hora de diseñar las embarcaciones. Por ello, se diseña con Rhinoceros y luego se importa a MaxSurf.

El resultado del diseño de la forma del casco de la embarcación es el siguiente:

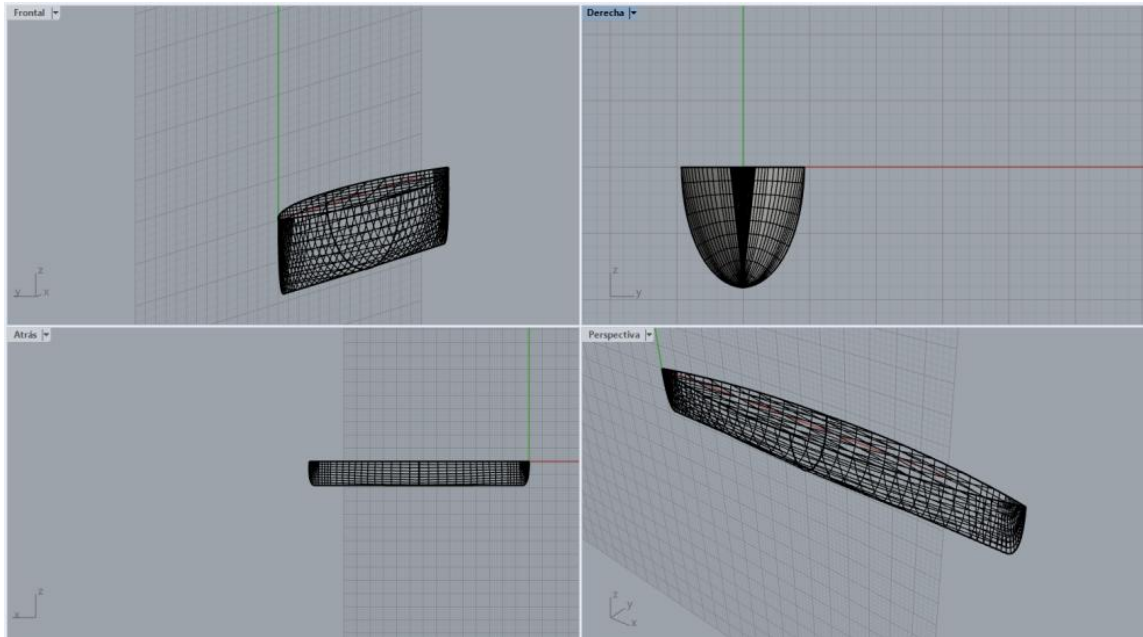


Figura 8. Pala del casco en Rhinoceros

La estructura ha sido realizada a base de curvas entrelazadas, esto permite crear un objeto de polilíneas cerrado, y así poder aplicar una malla, que más tarde nos facilitaría la impresión en 3D.

Debido a las dimensiones de las impresoras 3D, se tuvo que seccionar la estructura del casco en 6 partes para que se pudiesen imprimir sin ningún tipo de problema.

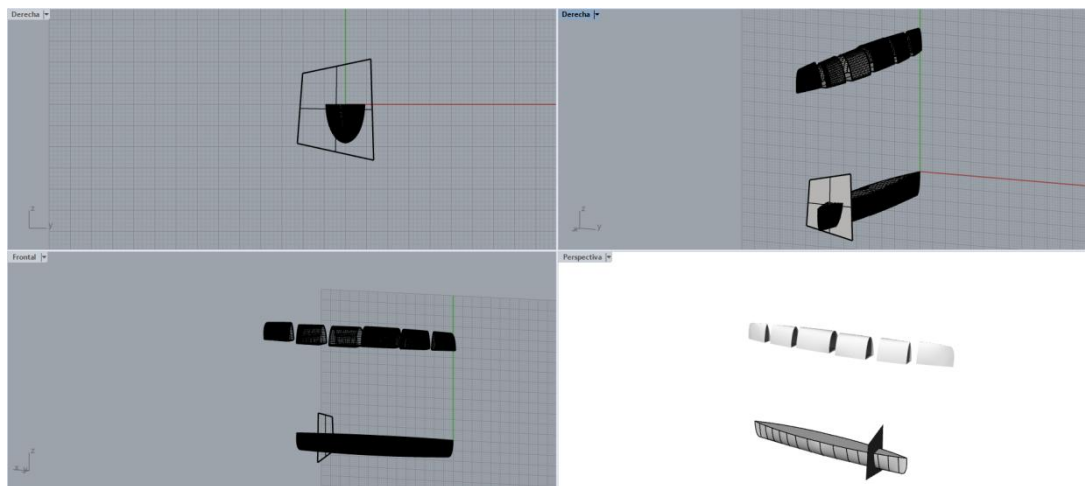


Figura 9. Casco seccionado para la impresión



También hay que tener en cuenta que, la estructura debe enfocarse hacia la placa para que el proceso de impresión se realice correctamente.

### 3.3 Maxsurf

Antes de imprimir, se ha exportado el documento de Rhinoceros a MaxSurf para corroborar que la estabilidad de las palas es la adecuada para navegar. Debido a la dificultad de importar el diseño del buque, ya que no se transfería por completo, se ha decidido recrearlo aproximadamente a partir de superficies de medio cilindro. Una vez se tiene el diseño de las palas, se comprueba sus capacidades mediante la acción de *Calculate Hydrostatics*, en esta nos permite observar todo tipo de características a cerca de la embarcación, como el calado o el centro de gravedad de la pala.

El diseño y resultados, son los siguientes:

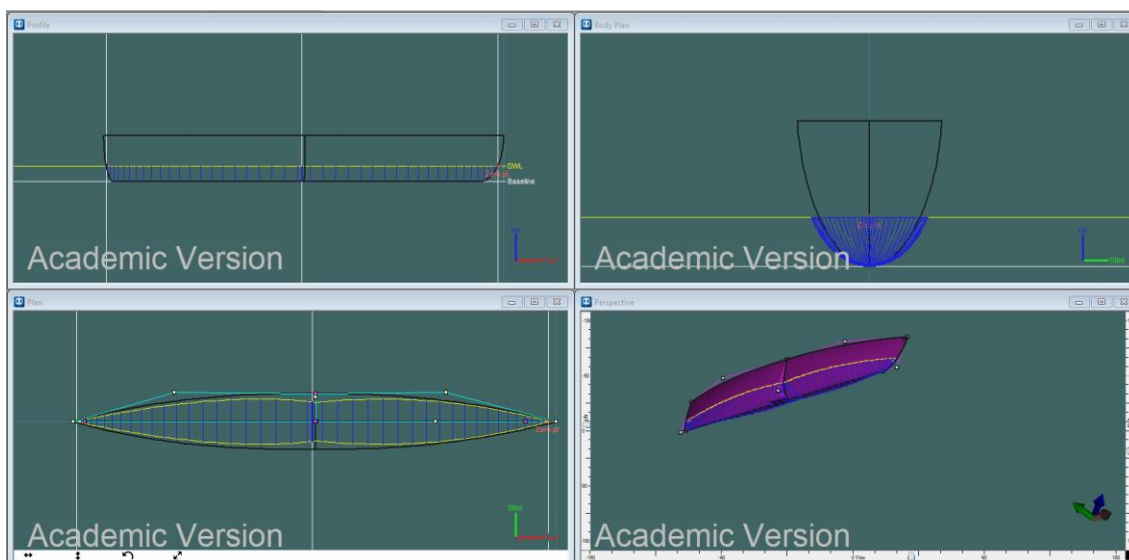


Figura 10. Prueba de flotabilidad en Maxsurf

Como se puede observar en la imagen, el calado teórico de la pala es el adecuado para navegar. Al tener tanta flotabilidad como relación directa al calado, cuando se añada una cantidad de pesos elevada, no habrá ningún incidente de hundimiento.

Además, los valores del radio metacéntrico o del centro de gravedad pasarán a ser más estables cuando se unan las dos palas con una estructura de madera, imposibilitando la escora con un gran ángulo.

	Measurement	Value	Units
1	Displacement	1,070	kg
2	Volume (displaced)	1043,95	cm <sup>3</sup>
3	Draft Amidships	3,26	cm
4	Immersed depth	3,26	cm
5	WL Length	81,21	cm
6	Beam max extents on	7,73	cm
7	Wetted Area	747,70	cm <sup>2</sup>
8	Max sect. area	17,98	cm <sup>2</sup>
9	Waterpl. Area	461,13	cm <sup>2</sup>
10	Prismatic coeff. (Cp)	0,715	
11	Block coeff. (Cb)	0,509	
12	Max Sect. area coeff.	0,713	
13	Waterpl. area coeff. (C)	0,734	
14	LCB length	-39,52	from ze
15	LCF length	-39,83	from ze
16	LCB %	-48,660	from ze
17	LCF %	-49,045	from ze
18	KB	1,94	cm
19	KG fluid	3,26	cm
20	BMT	1,67	cm
21	BML	159,65	cm
22	GMT corrected	0,34	cm
23	GML	158,32	cm
24	KMT	3,61	cm
25	KML	161,59	cm
26	Immersion (TPC)	0,000	tonne/c
27	MTc	0,000	tonne.m
28	RM at 1deg = GMT.Dis	0,01	kg.cm
29	Length:Beam ratio	10,502	
30	Beam:Draft ratio	2,369	
31	Length:Vol <sup>0.333</sup> rati	8,006	
32	Precision	Medium	73 stati

Tabla 3. Valores de flotabilidad del casco

### 3.4 Impresión

Después de todo el proceso de diseño del casco y de los cálculos teóricos de flotabilidad, ya podemos proceder a la impresión de la embarcación, para ello, son necesarios los datos de impresión para darle a la estructura la calidad que queramos, ya que según la densidad dentro de la estructura o el diámetro del filamento, obtendremos unos resultados u otros.

Los valores que se consideran óptimos son los siguientes:

Características de la impresora	
Nozzle diameter	0,4 mm
Características del filamento	
Diámetro filamento	1,75 mm
Temp. Extrusor 1ª capa	210 °C
Temp. Base 1ª capa	65°C
Temp. Extrusor	205°C
Temp. Base	65°C
Características impresión del sólido	
Capas interiores (Densidad)	0%
Patrón de capa	Rectilíneo
Patrón capas inferiores/superiores	Rectilíneo
Nº mínimo perímetros	3
Ángulo patrón	45º
Nº Capas sólidas superiores	3
Nº Capas sólidas inferiores	3
Velocidades de impresión	
Perímetros	30 mm/s
Perímetros pequeños	30 mm/s
Capas	30 mm/s
Capas sólidas	30 mm/s
Capas superiores	25 mm/s
Soportes	25 mm/s
Puentes	40 mm/s

Tabla 4. Datos de impresión

Cuando se apliquen todos los datos a la impresora, se podrá saber con exactitud la cantidad de tiempo y material que se requerirán.

DESCRIPCIÓN	CANTIDAD	PRECIO UNIDAD	SUBTOTAL	DTO.	TOTAL
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	9,7000	3,7200	36,08		36,08
FILAMENT PLA (gr)	71,0000	0,0450	3,20		3,20
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	17,0000	3,7200	63,24		63,24
FILAMENT PLA (gr)	121,0000	0,0450	5,45		5,45
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	27,0000	3,7200	100,44		100,44
FILAMENT PLA (gr)	192,0000	0,0450	8,64		8,64
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	24,0000	3,7200	89,28		89,28
FILAMENT PLA (gr)	166,0000	0,0450	7,47		7,47
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	18,0000	3,7200	66,96		66,96
FILAMENT PLA (gr)	127,0000	0,0450	5,72		5,72
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	12,5000	3,7200	46,50		46,50
FILAMENT PLA (gr)	93,0000	0,0450	4,19		4,19
TÈCNIC MANIP.	4,0000	9,0900	36,36		36,36

Tabla 5. Cantidad de tiempo y material

En la impresión de los 12 módulos que forman el buque se invertirán 216 horas durante dos semanas, de las cuales 54 horas se emplean en las piezas más grandes del casco. Inicialmente, estaba pensado llevarlas a cabo en 3 semanas, pero se redujo el tiempo al dejar las impresoras 3D encendidas durante la noche.

La cantidad total de filamento necesario para completar el casco son 800 gramos, con un precio de 69,34€ solamente en material.

El resultado final de la impresión de los diferentes módulos de PLA es el siguiente:



Figura 11. Módulos por separado

## Capítulo 4: Ensamblaje del casco

Después de todo el proceso de impresión, obtenemos 12 piezas de PVC independientes con una resistencia estructural muy baja. Para aportarle las características necesarias al casco, las piezas de PVC deben someterse a un proceso de masillado y pintura.

### 4.1 Unión de las piezas del casco

Una vez se tienen las piezas impresas, se efectúa la unión de la estructura, para ello se utilizarán los siguientes productos:

- Pegamento Bicomponente
- Masilla de poliéster
- Imprimación
- Pintura

#### 4.1.1 Pegamento Bicomponente

Este pegamento está formado por dos componentes, que al mezclarlos reaccionan entre sí con la intención de aportar una fuerte adherencia a los módulos. Un ejemplo, como referencia para este tipo de adhesivos, es el Epoxy.

Para realizar esta parte del proceso es aconsejable protegernos a nosotros mismos mediante guantes, gafas, bata y mascarilla de protección, ya que los componentes que se van a utilizar pueden dañarnos ya sea por el tacto o por la inhalación de los gases que desprende. Esta resina en particular, no es altamente peligrosa pero hay que tenerlo en cuenta a lo largo del proceso para que no suceda ningún accidente.

Teniendo en cuenta los factores anteriormente nombrados, se procede a la unión de la pieza. Primero, se mezclan los dos componentes con la misma cantidad y se aplican mediante una espátula, se requieren 5 minutos para aplicar la mezcla antes de que se endurezca. Al emplearla, hay que mantener unidas las piezas durante 20 minutos para obtener la rigidez y resistencia estructural necesaria. Este proceso se repite las veces que se requieran. En este caso, se aplicará 10 veces para unir todos y cada uno de los módulos.



Figura 12. Palas unidas por pegamento bicomponente

El bicomponente aplicado ha sido de la marca Ceys, este producto posee dos recipientes con 75 ml para cada elemento y una espátula de plástico que facilita la mezcla de los componentes.



Figura 13. Pegamento Bicomponente

Tanto la cantidad de material como las propiedades que aporta el pegamento a la embarcación han sido óptimas, ha servido para ensamblar todos los dispositivos y elementos en el catamarán.

#### 4.1.2 Masilla de poliéster

La masilla de poliéster se aplica con la finalidad de mejorar las imperfecciones de la impresión, en la superficie se puede comprobar que la capacidad de detalle de las impresoras 3D es buena, pero no lo suficiente.

En ciertas zonas, las capas de PVC son escasas y es necesario un extra de dureza, esta la añadiremos mediante capas de masilla fina de poliéster, que además de aportarnos la característica anteriormente nombrada, facilita el lijado de la superficie.

En primer lugar, se debe mezclar el catalizador con la masilla de relleno, aproximadamente 100 gramos de masilla de relleno con 2-3 gramos de endurecedor, para que sea más exacto el proceso se puede precisar mediante básculas, en caso contrario, la unión se puede realizar por volumen. El tiempo de mezcla no está determinado, pero no es recomendable por experiencia propia estar durante más de 2 minutos, ya que la masa resultante se endurecerá y no será útil, los módulos se unirán escasamente y la resistencia estructural será mínima.



Figura 14. Masilla de poliéster sin lijar

En este caso, al utilizar un material tan peligroso, es obligatorio utilizar elementos de protección como son los guantes, gafas, mascarilla y bata de protección ya que la inhalación de gases o el contacto del material con la piel puede generar graves molestias en el manipulador.

Una vez efectuada la mezcla, se aplica en toda la estructura para darle la fuerza necesaria que la resina de epoxy no puede aportar. Después de 20 minutos de haber aplicado la combinación, se reduce mediante lijado las zonas en las que haya exceso de material. Por otra parte, en las zonas donde haya imperfecciones, se puede volver a aplicar el proceso de añadido de masilla.

Posteriormente, la fase de lijar se realizará con tres tipos de grano. Inicialmente se utilizará un grano de 100 para rebajar la capa de masilla, ya que es la más gruesa y facilita la reducción de exceso de material, la segunda fase se efectuará con grano 240, que es menos gruesa, y la fase final, que requiere de más precisión, se utiliza grano 500 porque al ser tan fina, permite un acabado con un gran nivel de detalle.





Figura 15. Casco con la masilla de poliéster lijada

El producto que se ha aplicado, ha sido masilla fina de poliéster de la marca VossChemie. La cantidad que se encuentra dentro de los recipientes son 970g de masilla de relleno con 30g endurecedor. El resultado después de haberlo aplicado ha sido bastante satisfactorio, pero la peligrosidad de este material es bastante elevada, en caso de utilizarlo es muy recomendable utilizar guantes y sobre todo mascarilla, ya que los vapores que desprende son muy perjudiciales.



Figura 16. Masilla fina de poliéster

#### 4.1.3 Imprimación

La imprimación es un proceso de preparación de superficies que se utiliza en el casco para facilitar la adherencia de la pintura. En este caso, se aplicará en forma de spray y tiene como componentes la acetona y el acetato de butilo, hay que tener precaución a la hora de manipular este producto ya que es altamente inflamable, como en los anteriores procesos, se utilizará como material de protección: bata, guantes, gafas de protección y una mascarilla.





Figura 17. Imprimación multusos

El modo de empleo es muy sencillo, hay que aplicar la imprimación en toda la superficie homogéneamente. El tiempo de secado es de 30 minutos, como es tan bajo no hay problemas en dividir el proceso en dos partes, la primera se aplicará en los costados, y en la segunda se aplicará en la zona superior, por lo que para emplearlo en toda la embarcación serán necesarios 60 minutos. Este tipo de imprimación posee una gran lijabilidad por lo que, si hay alguna que otra gota en la superficie, se pueden lijar. Si queremos obtener un óptimo resultado, se recomienda lijar al agua con grano 600.



Figura 18. Casco del buque con la imprimación

#### 4.1.4 Pintura

La pintura aplicada en el casco es de esmalte uretanado especializada en ambientes marinos y climas extremos, procedente de la marca Titan Yate. Se emplea esta pintura no solo para mejorar estéticamente el buque, también sirve para protegerlo de la corrosión, ya que todo tipo de material en contacto con el agua marina es susceptible de sufrir este ataque electroquímico.



Figura 19. Pintura Titan Yate

El modo de empleo consiste en aplicar la pintura mediante pinceles, ya que no existe la variante en spray. Como no se puede emplear el esmalte en toda la estructura simultáneamente, al igual que en la imprimación, el proceso se divide en dos partes, la primera se destinaría en los costados, y la segunda, en la zona superior, de esta manera todo el catamarán queda pintado sin el peligro de que aparezcan imperfecciones.



Figura 20. Casco pintado

El tiempo necesario para que el acabado sea óptimo es de 24 horas, por lo que para efectuar todo el proceso de pintado de las dos palas, se requieren 48 horas al dividir el proceso en dos partes.

Como recomendación, una vez se haya efectuado todo el procedimiento de pintado, o solamente una parte, es preferible situar las piezas dentro de algún recinto en el que la climatología durante el día y la noche no pueda afectar a las piezas en cuestión.

El material de protección aconsejado para efectuar esta actividad es: una bata, guantes, gafas de protección y mascarilla.

## 4.2 Unión de las dos palas

Una vez se tiene todo el casco completo, se une mediante una estructura que aporte estabilidad, además de superficie para situar la caja con el circuito Arduino.

El material que se ha seleccionado para conectar las dos palas, ha sido la madera, ya que visualmente se asemeja al catamarán clásico, al ser robusto otorga resistencia estructural y es excelente para resistir esfuerzos con su debido tratamiento.

Aparte de soportar la estructura donde se situará el circuito Arduino, también servirá como apoyo para el sensor de ultrasonidos en la zona de proa y sujeción para los arbotantes en la zona de popa.

Las dimensiones de cada pieza de madera son las siguientes:

	Distancia (centímetros)
Longitud	38
Amplitud	4,7
Altura	1,1

Tabla 6. Dimensiones de los soportes

La unión de las piezas de madera con el casco se ha efectuado mediante resina epoxy, utilizando el mismo pegamento bicomponente que se empleó en el ensamblaje de los módulos. Las piezas se han situado lo más cerca posible del centro de la embarcación, con una desviación a popa con la intención de mejorar el rendimiento de las hélices.



Figura 21. Unión de las palas con las piezas de madera

### 4.3 Caja

Debido a la sensibilidad de los circuitos electrónicos en la mar, es indispensable protegerlo para no sufrir ningún daño, para ello se ha construido una estructura de PVC. Se ha elegido este material por su bajo peso en relación al volumen que ocupa.

Las dimensiones son las siguientes:

	Distancia (centímetros)
Longitud	38,5
Amplitud	39
Altura	10,5

Tabla 7. Dimensiones de la caja

Se ha situado ligeramente a popa al igual que las piezas de madera, para crear un trimado apopado, de esta manera los motores tendrán un mejor ángulo de ataque y generarán un mayor impulso, exactamente se sitúa 2,5 centímetros más apopado desde el centro de geométrico de la embarcación.

Para su creación, se utilizó una placa de PVC de aproximadamente 1,60 x 0,60 metros, que se recortó en dos placas de 38,5 x 39 centímetros para la parte superior e inferior, dos placas de 38,5 x 10,5 centímetros para el costado y finalmente, dos placas de 39 x 10,5 centímetros para el mismo propósito que las anteriores, el costado.

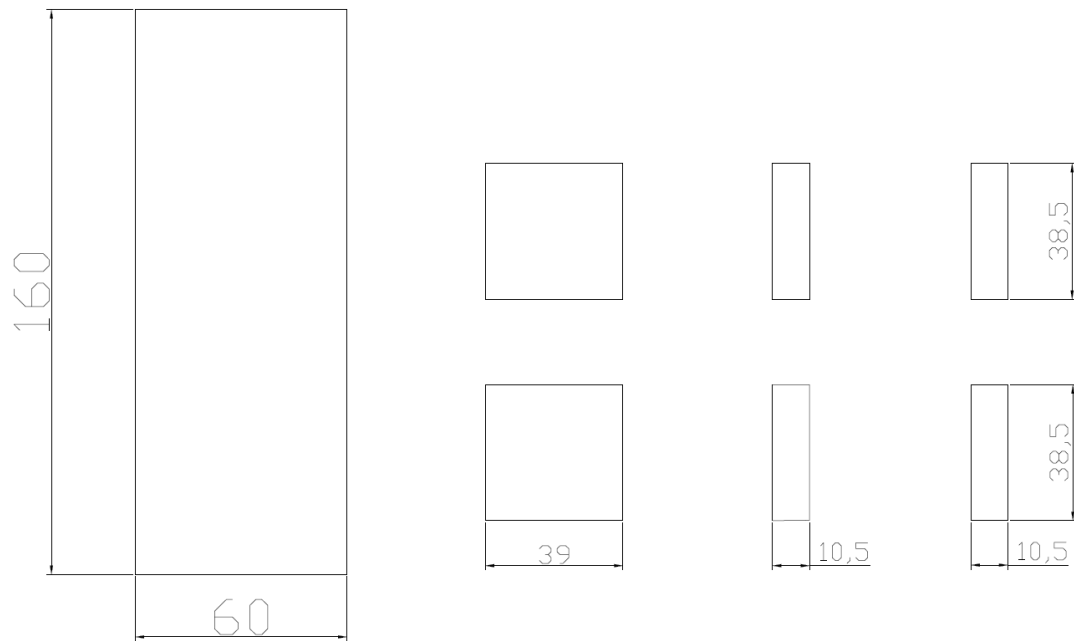


Figura 22. Medidas de la placa de PVC

Una vez se ha recortado la placa de PVC, se unen los fragmentos mediante perfiles en L. La vara de PVC medía 2,5 metros y se recortó en diferentes longitudes según la necesidad.

La unión de las placas con los perfiles en L se realizó mediante resina Epoxy, de la misma manera se unió las piezas de madera a las palas que forman el casco del buque. Los resultados de aplicar este adherente, son muy satisfactorios.



Figura 23. Caja

#### 4.4 Arbotantes

Una de las piezas más importantes en los buques es el arbotante, esta pieza se encarga de sostener los ejes de las hélices para realizar un impulso con la mayor eficacia posible.

Debido a que el presupuesto de la embarcación se elevaba demasiado, se eliminó la posibilidad de realizarlos mediante impresión 3D. Por lo que se barajó la posibilidad de elaborarlos con madera, como se hizo con los elementos de sujeción de la caja, de esta manera, el buque mantiene un aspecto rústico.

Los arbotantes, se realizaron en dos procedimientos, primero se utilizó una pieza de madera rectangular de 1.5x1.5x3 centímetros, que posteriormente se taladró por el centro con la intención de sujetar los ejes de transmisión. Por otra parte, se unió a esta pieza taladrada, una cuña de madera que inclinaba la primera, con el objetivo de realizar el impulso de forma óptima. Aunque, al principio la idea no parecía muy efectiva, una vez concluida, los resultados fueron excelentes.

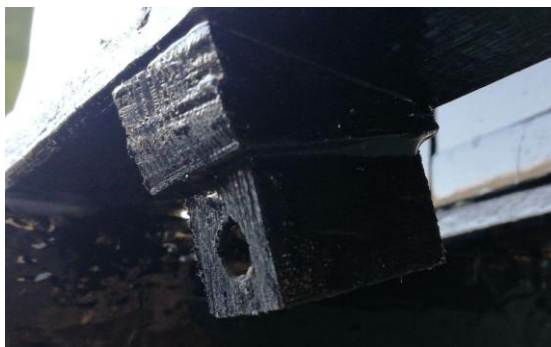


Figura 24. Arbotante

## Capítulo 5: Instrumentación y control

Todos los procesos en el desarrollo del buque son importantes, pero el más relevante con diferencia, es la creación del circuito encargado de realizar maniobras con la intención de no impactar con ningún obstáculo.

Para saber que componentes se requieren para el circuito, hace falta comprender cómo queremos enfocarlo, dependiendo de la intención, escogeremos unos elementos u otros.

### 5.1 Elementos para obtener información del medio

En el mar, los objetos que se sitúan alrededor del buque varían de posición constantemente, hay que tener presente la posición de estos, para que no exista posibilidad de colisión. Los dispositivos que desarrollan la tarea de localizar objetos cercanos al catamarán son:

- Sensores infrarrojos
- Sensores de ultrasonidos

#### 5.1.1 Sensores infrarrojos

Este elemento es un dispositivo capaz de detectar la presencia de un objeto mediante la reflexión que produce en la luz. Los sensores de infrarrojos están formados por un LED emisor de luz infrarroja y de un fotodiodo (LM393) que recibe la luz reflejada por un posible obstáculo.[7]

Los detectores de obstáculo suelen proporcionarse con una placa de medición estándar con el comparador LM393, que permite obtener la lectura como un valor digital cuando se supera un cierto umbral, que se regula a través de un potenciómetro ubicado en la placa.

Este tipo de sensores actúan a distancias cortas, entre 5 y 20mm. Además la cantidad de luz infrarroja recibida depende del color, material, forma y posición del obstáculo, por lo que no disponen de una precisión suficiente para proporcionar una estimación de la distancia al obstáculo.

Pese a esta limitación son ampliamente utilizados para la detección de obstáculos en pequeños vehículos o robots. Su bajo coste hace que sea frecuente ubicarlos en el perímetro, de forma que detectemos obstáculos en varias direcciones.

### 5.1.2 Sensores de ultrasonidos

Un sensor de ultrasonidos es un dispositivo para medir distancias. Su funcionamiento se basa en el envío de un pulso de alta frecuencia, no audible por el ser humano. Este pulso rebota en los objetos cercanos y es reflejado hacia el sensor, que dispone de un elemento adecuado para esa frecuencia.[8]

Midiendo el tiempo entre pulsos, conociendo la velocidad del sonido, podemos estimar la distancia del objeto contra cuya superficie impacta el impulso de ultrasonidos

Los sensores de ultrasonidos son sensores baratos, y sencillos de usar. El rango de medición teórico del sensor HC-SR04 es de 2cm a 500 cm, con una resolución de 0.3cm.

Los sensores de ultrasonidos son sensores de baja precisión. La orientación de la superficie a medir puede provocar que la onda se refleje, falseando la medición.



Figura 25. Sensor de Ultrasonidos Hc-SR04

Factores como la alta velocidad de navegación y el bajo coeficiente de fregamiento en el agua, son cruciales a la hora evitar colisiones. Teniendo en cuenta los argumentos anteriormente nombrados, los sensores de ultrasonidos se asemejan a los requisitos aportando un alto tiempo de reacción, eliminándose como opción los sensores de infrarrojo.



## 5.2 Elementos para administrar la información

Todo circuito electrónico necesita una placa central que sea capaz de administrar toda la información que recibe de los sensores, estos se ubican alrededor del buque, para aumentar el rango de superficie a la que es capaz de reaccionar.

Existen diferentes dispositivos que administran la información, que puede ser aptos para este proyecto.

### 5.2.1 Arduino

Es una plataforma de creación de prototipos de código abierto basada en *hardware* y *software* libre y fácil de usar, una vez se escribe el código, se puede subir a la tarjeta a través de un Entorno de Desarrollo Integrado. [10]

A través de los años hasta este 2018, esta plataforma ha impulsado miles de proyectos. Ha crecido en torno a una comunidad donde principiantes y expertos de todo el mundo comparten ideas, conocimientos y su experiencia colectiva.

### 5.2.2 Raspberry Pi

Es un miniordenador con el tamaño de una tarjeta de crédito, bastante barato (menos que Arduino), desarrollada en Reino Unido por la Fundación Raspberry Pi, con el objetivo principal de estimular la enseñanza de ciencias de la computación en las escuelas. Al ser un miniordenador podemos utilizarlo para desarrollar cosas bastante más complejas que con Arduino y puede utilizar lenguajes de programación de alto nivel como Python, C++ y Java. [24]

Ambas placas pueden servir para el proyecto, pero debido a la accesibilidad en puertos digitales y analógicos, y su simplicidad en la programación, la placa Arduino Uno es la seleccionada para desarrollar el trabajo.



Figura 26. Placa Arduino Uno

### 5.3 Código

La plataforma Arduino se programa con un lenguaje propio basado en el lenguaje de programación de alto nivel, similar a C++.

Se trata de un lenguaje de programación de propósito general asociado a un sistema operativo llamado UNIX. este lenguaje de alto nivel, trata con objetos básicos como caracteres, números, bits y direcciones de memoria, entre otros.

Este tipo de lenguaje posee una gran portabilidad. Gracias a ello se suele utilizar para la programación de sistemas como la construcción de intérpretes, compiladores, y editores de texto.

Todos los códigos desarrollados durante el trabajo se encuentran en el Anexo.

### 5.4 Elementos que actúan según el código

Los elementos del buque encargados de modificar sus acciones según el código, es el dispositivo de doble puente H y los dos motores.

Motores: Son los encargados de impulsar la embarcación, están alimentados por pilas de 9V a pesar de rendir óptimamente a los 12V, están conectados con ejes y hélices, y se sostienen mediante los arbotantes.

El tipo de motor, ha sido elegido, teniendo en cuenta el circuito que se planeaba construir, al querer alimentar todo el circuito con pilas de 9V, se optó por el motor que ofrecía más rev/min, este motor puede girar a 11500 rev/min y puede ser alimentado por 12-9 V.

Para este proyecto se ha requerido de dos motores, en vez de uno con un timón para variar la dirección de la nave. De esta manera, el proceso de ensamblaje se acorta, al igual que el presupuesto que se hubiese necesitado para crear esta pieza.



Figura 27. Motor de 12V

Doble puente H: El módulo cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, entre ellos diodos de protección y un regulador **LM7805** que suministra 5V a la parte lógica del integrado L298N. Cuenta con jumpers de selección para habilitar cada una de las salidas del módulo.

La intención de utilizar este módulo en el trabajo es dar las órdenes de la placa a los motores y permite que ambos funcionen independientemente, según la información que facilite la placa Arduino. De esta manera, al utilizar cada motor según nuestras necesidades, se puede girar el buque más fácilmente, invirtiendo solamente un motor o incluso frenar, invirtiendo los dos motores.

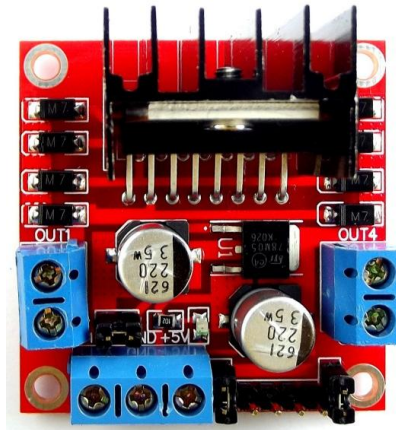


Figura 28. Dispositivo doble puente H L298N

## Capítulo 6: Plan de pruebas y resultados

A lo largo del trabajo final de grado se han realizado diversas pruebas, para hacernos una idea de la capacidad que tiene la embarcación para evitar obstáculos.

### 6.1 Prueba de flotabilidad

La primera prueba a la que debe enfrentarse la embarcación, es a la flotabilidad. A pesar de haber realizado esta prueba anteriormente en el programa de diseño MaxSurf y haberla superado con éxito, aún no había constancia de la efectividad del buque en el agua.

El desarrollo de la prueba es sencillo, simplemente hay que colocar el catamarán en la piscina, para tomar notas a cerca del calado, relación peso-calado, altura de la posición de las hélices y distribución de pesos dentro de la caja, que puede derivar en la variación de la dirección del barco.



Figura 29. Prueba de flotabilidad con casco

Esta prueba se desarrolló en dos partes, primero con la estructura del barco, como se puede observar en la Figura 17, y luego, añadiendo la caja con los ejes de transmisión, cardans y hélices.

Inicialmente con la estructura del barco, se puede observar que la altura del calado representa un tercio del puntal, esto indica que la flotabilidad del barco es muy elevada y los resultados

en MaxSurf han sido acertados. Luego, al añadir la caja con los ejes de transmisión, cardans y hélices, tuvo como consecuencia una variación en la distribución de pesos, que aumentó el calado hasta la mitad del puntal.

Finalmente se añadieron diversos pesos para analizar el comportamiento del modelo, según aumentaba la carga, el francobordo disminuía en forma de exponencial, esto muestra que el modelo se opone con excelentes resultados y mantiene un buen nivel de calado.

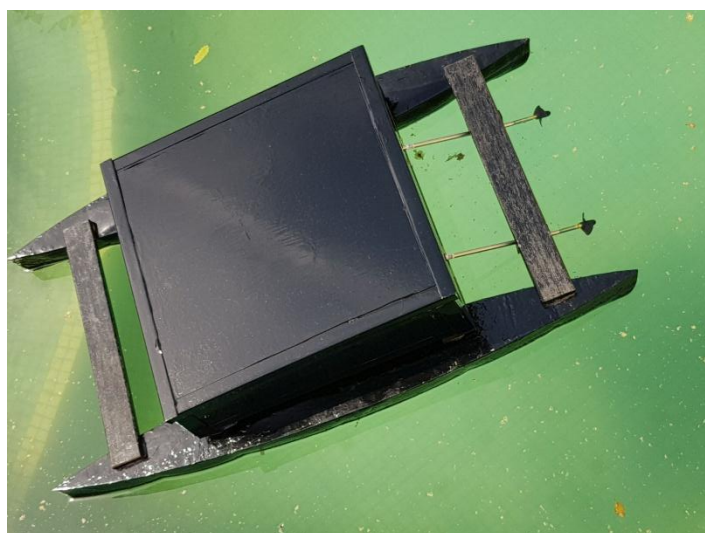


Figura 30. Prueba de flotabilidad con casco y caja

## 6.2 Prueba de Ultrasonidos

El sensor de ultrasonidos es la pieza principal en la que se basa todo el mecanismo de detección y evasión de obstáculos, con la intención de aplicarlo óptimamente al buque, se recopila toda la información posible a cerca de sus propiedades.

Al buscar este dispositivo en Google, se puede contemplar que el rango de visibilidad del ultrasonidos varía de 2 a 500 cm, para corroborar la información facilitada en internet, se ha realizado un código Arduino que permite observar la longitud de detección de objetos del Dispositivo HC-SR04.

```

sketch_feb22a
#define Pecho 4
#define Ptrig 3
long duracion, distancia;

void setup() {
  Serial.begin (9600);
  pinMode(Pecho, INPUT);
  pinMode(Ptrig, OUTPUT);
  pinMode(13,1);
}

void loop() {
  digitalWrite(Ptrig, LOW);
  delayMicroseconds(2);
  digitalWrite(Ptrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Ptrig, LOW);

  duracion = pulseIn(Pecho, HIGH);
  distancia = (duracion/2) / 29;

  if (distancia >= 500 || distancia <= 0){
    Serial.println("---");
  }
  else {
    Serial.print(distancia);
    Serial.println("cm");
    digitalWrite(13,0);
  }
  if (distancia <= 10 && distancia >= 1){
    digitalWrite(13, 1);
    Serial.println("Alarma.....");
  }
  delay(400);
}

```

Figura 31. Código de ultrasonidos

Durante la programación del código hay que tener en cuenta que la distancia de referencia se debe dividir entre dos, ya que la onda recorre dos veces el mismo trayecto, y que la velocidad del sonido es 1/29 cm/us.

$$D = \frac{\left(\frac{d}{2}\right)}{29} \quad (4)$$

Donde  $D$  es la distancia en centímetros y  $d$  la duración en microsegundos.

Además, cada pulso tarda 10 microsegundos en llevarse a cabo.

Subiendo el código a la placa Arduino, se logró obtener como distancia máxima 460 cm.

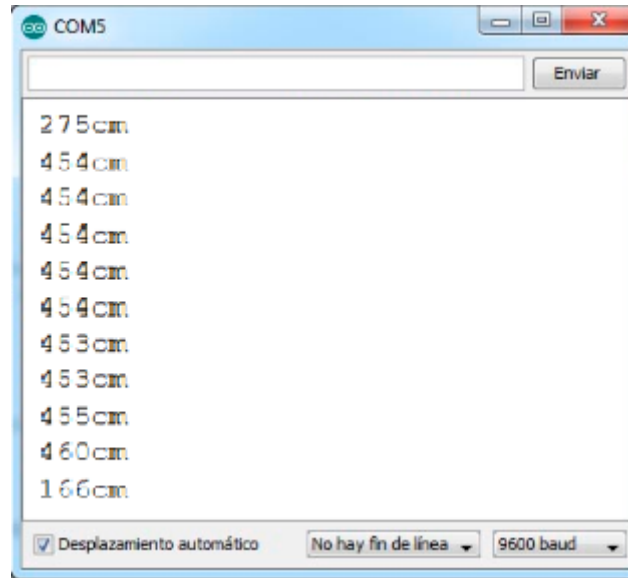


Figura 32. Valores máximos de la prueba de ultrasonidos

En cambio, la distancia mínima a la que es capaz de detectar el sensor corresponde a 4 cm

```
5cm
Alarma.....
4cm
Alarma.....
5cm
Alarma.....
10cm
Alarma.....
```

Figura 33. Valores mínimos de la prueba de ultrasonidos

Teniendo en cuenta los resultados, no habrá ningún problema en localizar obstáculos a largas distancias, ya que los valores teóricos se asemejan bastantes a los prácticos.

	Teórico	Práctico
Valor mínimo	2 cm	4 cm
Valor máximo	500 cm	460 cm

Tabla 8. Comparación teórico-práctica de los sensores de ultrasonidos



### 6.3 Prueba de freno

Las embarcaciones en el mar no se comportan de la misma manera que un automóvil en tierra, como el agua posee una diferente estructura molecular al ser un líquido, la fuerza de fricción creada por la oposición de las fuerzas, es diferente a la que sería creada en un terreno sólido como es el pavimento. Este es un factor a tener en cuenta durante todo el proyecto.

Inicialmente, esta prueba iba a dividirse en dos partes, la primera consistía en el frenado del buque invirtiendo el sentido de los motores y la segunda se basaba en la pérdida de la inercia de la embarcación a través de la inmovilización de los motores, es decir, sin funcionamiento. Pero debido a los malos resultados obtenidos durante la primera parte, se decidió solamente hacer pruebas invirtiendo los motores.

Primero, se situó el barco en el agua, el código que se subió a la placa Arduino, invertía los motores una vez los sensores de ultrasonidos detectasen un obstáculo a 1 metro. Como la piscina tenía 30 centímetros de altura de diferencia entre el agua y tierra, era una situación excelente para que los límites de la piscina representasen los obstáculos.

En el primer intento, el barco alcanzó tanta inercia que al invertir los motores a 1 metro, no era posible evitar la colisión con el borde de la piscina. Después de varios intentos, al variar la distancia de reacción a 3 metros, daba tiempo a que la embarcación frenase justo antes del choque con la pared, este fue el primer gran paso en el proyecto.

El código utilizado durante la prueba se encuentra en el Anexo (A.4. Freno con los dos motores invertidos.)

La dificultad de la programación se encuentra en definir todas las variables. El bucle que ejecuta la inversión de los motores es muy simple, ocupa solamente 20 líneas de código y es efectivo.

```
void loop() {  
  
    int distanceRight = 0;  
    int distanceLeft = 0;  
    delay(50);  
  
    if (distance <= 100) {  
        moveBackward();  
    }  
    else {  
        goesForward=false;  
        moveForward();  
    }  
    distance = readPing();  
    Serial.println(distance);  
}
```

Figura 34. Bucle loop

Durante la prueba, hubieron muchos fallos, unos ejemplos pueden ser la pérdida de una hélice y arandelas en el fondo de la piscina, que los motores no funcionasen correctamente, algunos elementos se sobrecalentaron...

Al finalizar, se observaron factores que influenciaron los resultados. El circuito Arduino al estar alimentado por pilas de 9V, obtenía de ellas una cantidad de energía que disminuía en el tiempo, por lo que la potencia desarrollada por el motor no era constante, así que los resultados obtenidos son aproximaciones de los comportamientos del catamarán.

Por otra parte, el viento y las corrientes en la piscina, alteraban la capacidad de freno y por consecuencia, los resultados.

## 6.4 Prueba de avance

La velocidad de la embarcación es uno de los aspectos a tener en cuenta en la capacidad de esquivar de obstáculo, esta variable viene definida por la forma del casco y la potencia de los motores entre otros factores.

El desarrollo de la prueba es muy sencillo, simplemente hay que conocer la distancia que va a recorrer el catamarán y simultáneamente a su movimiento, se va calculando el tiempo que tarda en realizarlo. El código es exacto al utilizado en la prueba de frenado. (Anexo A.4. Freno con los dos motores invertidos.)

Nuevamente, existe un inconveniente en la alimentación del circuito. Este contratiempo radica en la energía que son capaces de aportar las pilas, ya que disminuye según pasa el tiempo, por lo que la velocidad de avance de la embarcación, no es siempre la misma. Hay que considerar un margen de error en las pruebas que se realizan.

Durante las pruebas de avance que se efectuaron en la piscina de 5 metros de longitud, el catamarán fue capaz de realizarlos en 5 segundos, por lo que su velocidad media es de 1 m/s.

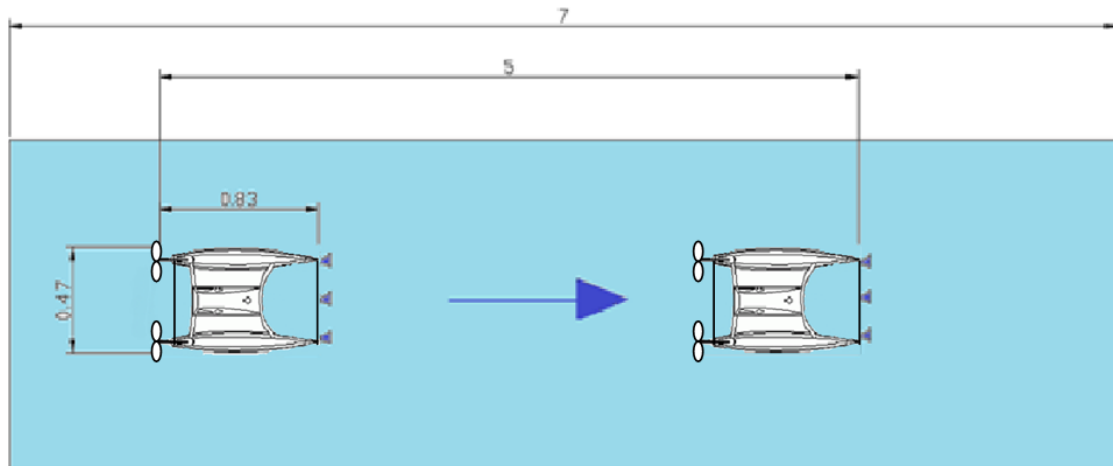


Figura 35. Representación del buque en la piscina durante la prueba de avance

## 6.5 Prueba de giro

Otra de las pruebas necesarias para analizar el comportamiento del buque, es la prueba de giro. Consiste en calcular el tiempo que tarda la embarcación en girar 90 grados.

Durante la práctica, se obtuvieron resultados muy distintos. Inicialmente, la embarcación lograba girar hasta los 90 grados, pero debido a la dirección de la corriente que había en la piscina, el rumbo del buque cambiaba, de manera que solo se podía conseguir como giro total, 90 grados.

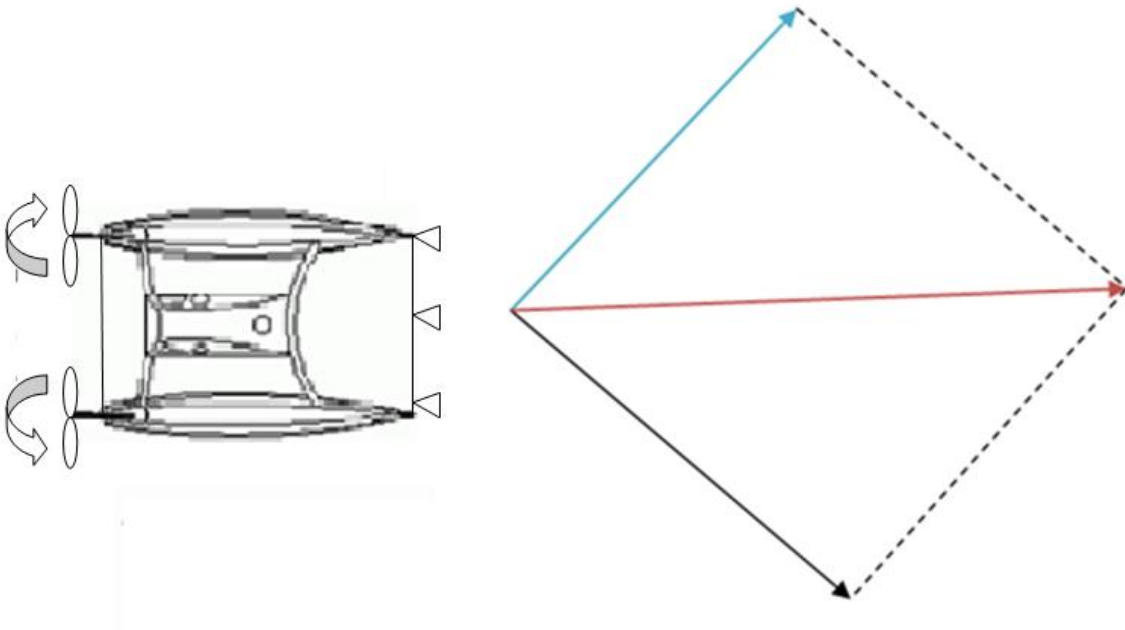


Figura 36. Comportamiento del buque con la corriente

La suma de fuerzas de los motores y la corriente resultaban en el avance de la embarcación sin capacidad de giro.

El código aplicado durante las prácticas mantiene la misma definición de variables, solamente varía el bucle y lo hace de la siguiente manera:

```
void loop(){

    int distanceRight = 0;
    int distanceLeft = 0;
    delay(50);

    if (distance >= 100){
        turnLeft();
    }
    else{
        goesForward=false;
        moveForward();
    }
    distance = readPing();
    Serial.println(distance);
}
```

Figura 37. Bucle en la prueba de giro

Se podía haber programado de muchas formas, pero elegí una en la que el buque giraba a babor siempre y cuando no detectasen nada los sensores a menos de 1 metro de distancia, al ser una piscina relativamente grande, no había ningún problema.

Una vez más, al igual que en las pruebas anteriores, la energía aportada por las pilas de 9V variaba el comportamiento de la embarcación. Según avanzaba el tiempo, el buque obtenía una capacidad de giro más elevada.

Finalmente, se logró que el catamarán fuese capaz de girar 90 grados sin ningún tipo de problemas, eso sí, después de horas de fallos en motores, calentamiento de los dispositivos del circuito y pérdida de elementos en la piscina.

## 6.6 Modificaciones

Inicialmente, se planteaba la idea de esquivar objetos mediante un sensor de ultrasonido, que sería capaz de detectar los elementos que se encontrasen a su paso, este se situaría en la zona de proa encima de un servomotor, de manera que al girar pudiese detectar un gran rango de superficie mientras avanza.

Debido a varios factores como la velocidad de la embarcación, el fluido en el que se encuentra, el tiempo de giro del servomotor y el tiempo de reacción del sensor, tienen como consecuencia la reducción de la efectividad en la localización de obstáculos. Por lo tanto, se ha decidido instalar tres sensores de ultrasonidos con la intención de disminuir la probabilidad de colisión.



Figura 38. Sensores de ultrasonidos instalados

De esta manera, mientras avance la embarcación, en cualquier momento, se pueden localizar los obstáculos y actuar al instante.

### 6.7 Prueba de evitar la colisión con el borde de la piscina

Debido a la complejidad del problema, se procedió solamente a detectar y evitar los bordes de la piscina, que al tener desnivel, los sensores de ultrasonidos, serían capaz de detectarlos.

Las dimensiones de la piscina son las siguientes:

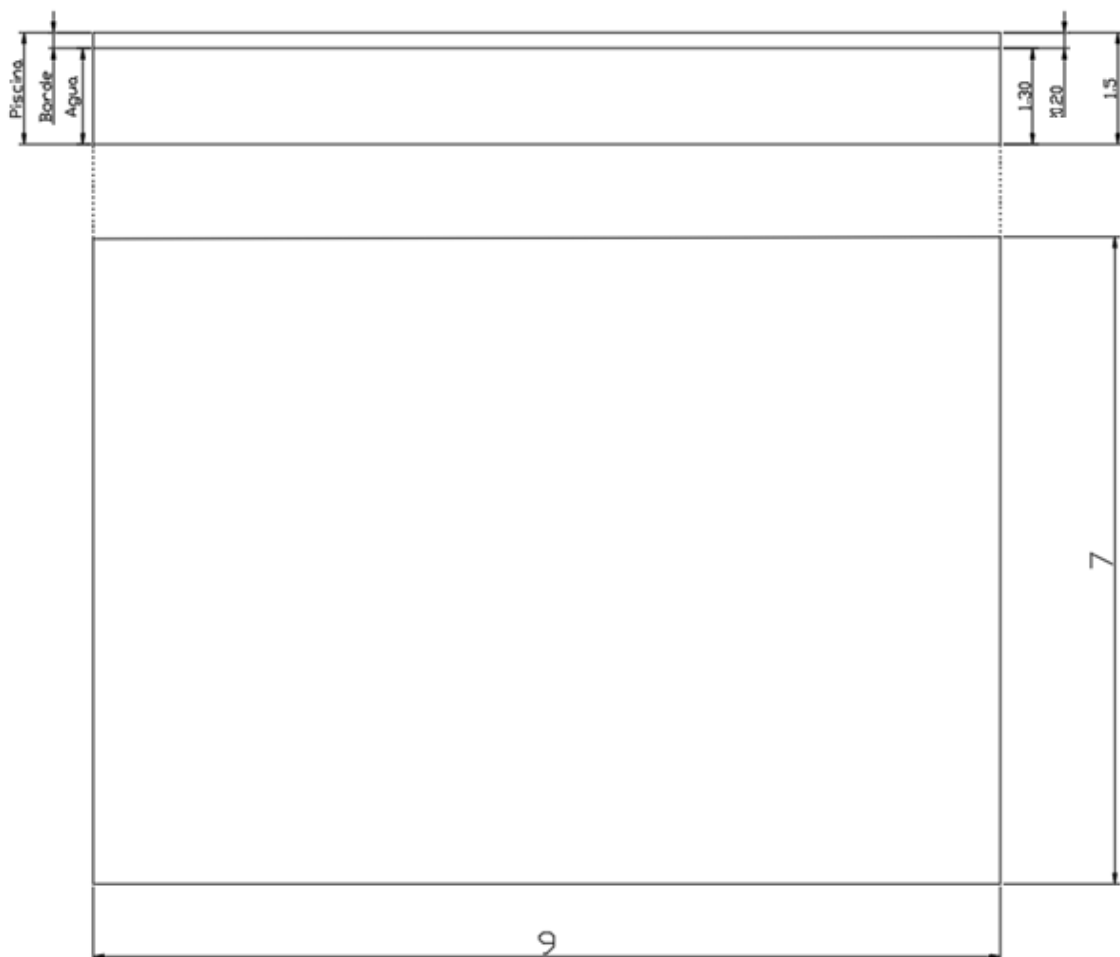


Figura 39. Dimensiones de la piscina con vista superior y frontal

Con la vista superior se puede observar que la piscina tiene 9 metros de longitud con 7 metros de anchura. Por otra parte, desde la vista frontal se puede ver que la altura total de la piscina

es de 1.5 metros, de este valor 1.3 metros son de agua, la distancia restante entre la superficie del agua y el borde de la piscina es un desnivel de 0.2 metros que sirve para que el ultrasonidos detecte la pared como obstáculo y la embarcación no colisione.

Las pruebas se realizaron en 3 piscinas diferentes, la superficie de cada una de ellas es similar, por lo que no hubieron problemas al cambiar de instalaciones.



Figura 40. Piscina número 1



Figura 41. Piscina número 2



Figura 42. Piscina número 3



El algoritmo que rige las acciones del catamarán es el siguiente:

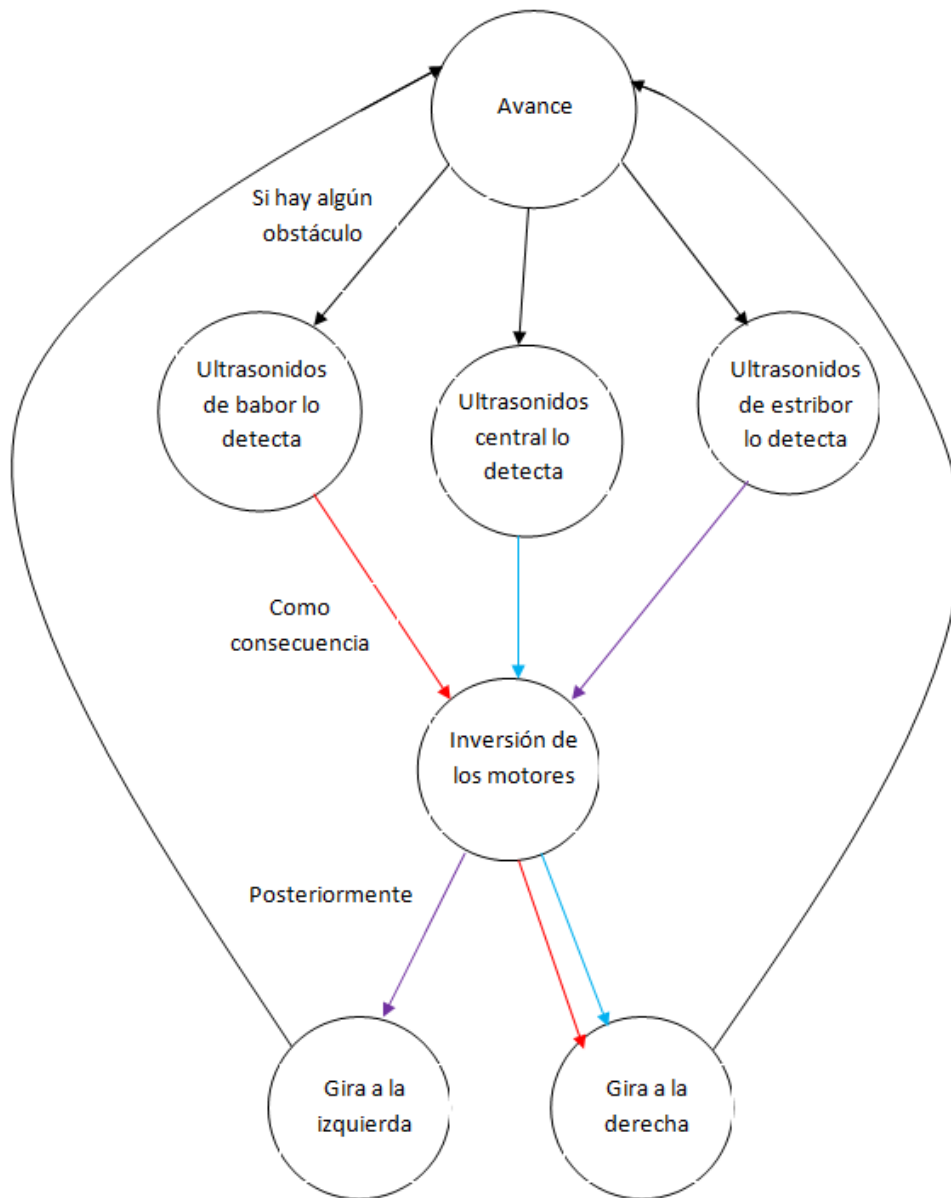


Figura 43. Diagrama de estados inicial

Se pretende que una vez empiece a avanzar el buque, cuando detecte un objeto, dependiendo del sensor de ultrasonidos que lo localice, se procederá de diferente manera. En los tres casos, se comienza con la inversión del sentido de ambos motores para frenar el catamarán, pero si el sensor que ha sido sorprendido se encuentra en el costado de babor, una vez se haya frenado

el buque, este invertirá el motor izquierdo y mantendrá la velocidad de avance del motor derecho. En cambio, para los otros dos sensores, variará el procedimiento, se invertirá el motor derecho y el izquierdo mantendrá el sentido de avance. Una vez ninguno de los sensores detecte algún obstáculo, se volverá al estado original del diagrama, es decir, el avance de la embarcación.

Durante esta prueba, hubieron muchos problemas, uno de ellos, era que los sensores de ultrasonidos al tener baja precisión, podían tomar malas referencias y actuar de forma errónea, afortunadamente, no sucedió con frecuencia, así que se pudieron captar tomas perfectamente.

Otro de los problemas, era que los cables, se desconectaban o incluso se cortaban de la presión que soportaban, de manera que, muchas de las pruebas se anularon debido a que los motores no funcionaban, por suerte, al ser un problema técnico tan simple, se pudo resolver de inmediato.

Una vez más, existía el problema de que la fuente de alimentación se agotaba a lo largo del tiempo, pero a diferencia de las otras pruebas, la energía que aportaban era similar, ya que, una gran cantidad del tiempo estaba siendo invertido en mejorar el código, además las pilas se fueron renovando cada cierto tiempo con la intención de que los resultados fuesen lo más exacto posible.

Después de realizar todos los arreglos para lograr que el código funcionase correctamente, se alcanzó la meta de no colisionar con ninguna pared

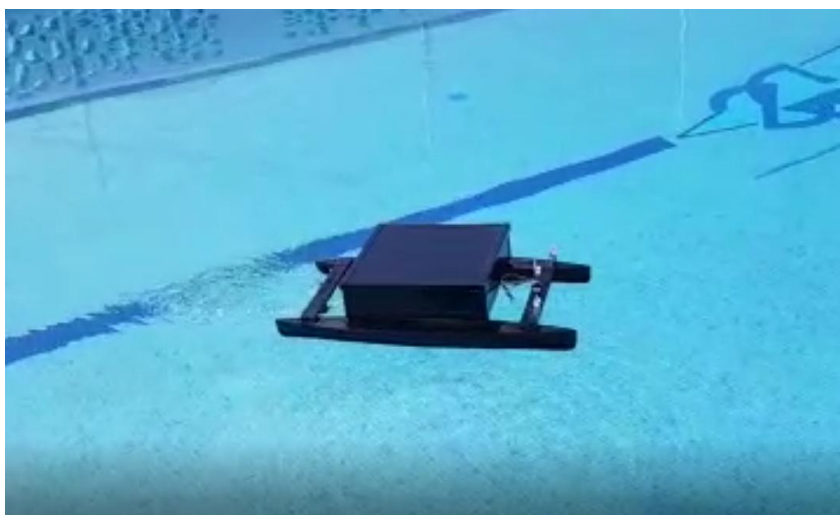


Figura 44. Comportamiento de la embarcación durante la prueba

El código utilizado durante la prueba, se encuentra en el ANEXO A.5. Código para evitar la colisión con el borde de la piscina.

A diferencia de los códigos utilizados en las otras pruebas, esta vez se utilizan 3 sensores de ultrasonidos y hay más variables a definir.

Durante la prueba, se puso de manifiesto que la capacidad de giro de la embarcación era muy reducida, para elevar este factor, era preciso que el buque tuviese una velocidad cercana a 0 m/s. Otra alternativa existente, era la posibilidad de instalar un timón, que por falta de presupuesto y de tiempo, no se ha llevado a cabo.

## 6.8 Prueba de llegar a un lugar y volver

Esta prueba consiste en navegar hasta un punto y lograr retornar hasta el punto original. El diagrama de estados que representa las acciones del código es el mismo que el de la Figura 29, solamente varía el tiempo de giro de las variables `turnLeft` y `turnRight`. Inicialmente, se programaron con la intención de que el buque pueda girar 90 grados, pero una vez se ejecute la acción, al querer alcanzar el objetivo de llegar y volver, la variación del ángulo sería el doble, 180 grados, por lo que se duplicó el tiempo de giro en el código de 9 a 18 segundos.

```

void turnRight() {

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorForward, LOW);

    delay(18000);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);

}

void turnLeft() {

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);

    delay(18000);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);

}

```

Figura 45. Variables definidas

## 6.9 Prueba de evitar la colisión con obstáculos

Una vez se ha logrado evitar la colisión con el borde de la piscina, se pretende alcanzar un objetivo mayor consiguiendo que el catamarán sea capaz de evitar obstáculos y continuar con la ruta previamente establecida. Teniendo en cuenta la dificultad, se aborda el problema mediante cálculos de tiempos a partir de los resultados en las pruebas de frenado, giro y avance.

El diagrama de estados es el siguiente:

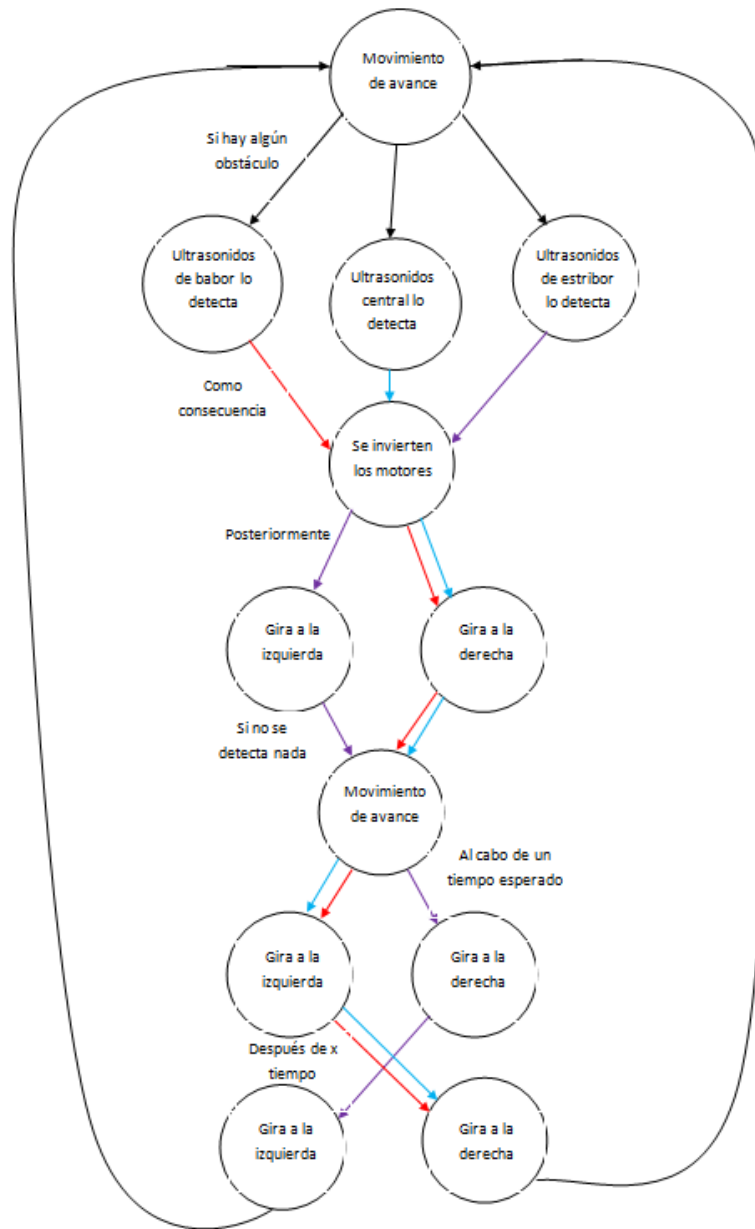


Figura 46. Diagrama de estados para evitar obstáculos

A diferencia del diagrama de estados de la Figura 28, este tiene cronometrados unos tiempos, que se calcularon en las pruebas de giro, avance y frenado. Gracias a estos valores, se sabe aproximadamente el tiempo necesario para evitar el obstáculo y volver a la ruta deseada.

Inicialmente, el buque avanza con una velocidad de 1 metro/segundo hasta que alguno de los sensores detecte un obstáculo a una distancia de 3 metros, según el sensor que los detecta, se invierte el sentido de los dos motores para inicialmente frenar el barco y posteriormente, se mantiene uno de los dos motores en sentido contrario, para poder abatirse al lado que se desee.

Seguidamente, el catamarán vuelve a efectuar la misma maniobra, frenando el barco invirtiendo el sentido de los dos motores y girando al lado opuesto que había girado inicialmente, en este punto, la nave se encontraría en el punto 10 de la Figura 32.

Finalmente, con la intención de volver a la ruta previamente establecida, se vuelve a invertir solamente un motor como pasó en el punto 4 de la Figura 32 y en el inicio de todo el proceso de esquivar.

La representación de todo el proceso, es el siguiente:

1. Avance de la embarcación.
2. El sensor de estribor detecta un obstáculo a menos de 3 metros.
3. Los motores invierten el sentido para frenar.
4. Inversión del motor de babor.
5. El barco se encuentra en la dirección correcta.
6. Avance de la embarcación.
7. Inversión del motor de estribor.
8. Posición correcta del buque.
9. Avance.
10. Inversión del motor de babor.
11. Avance.

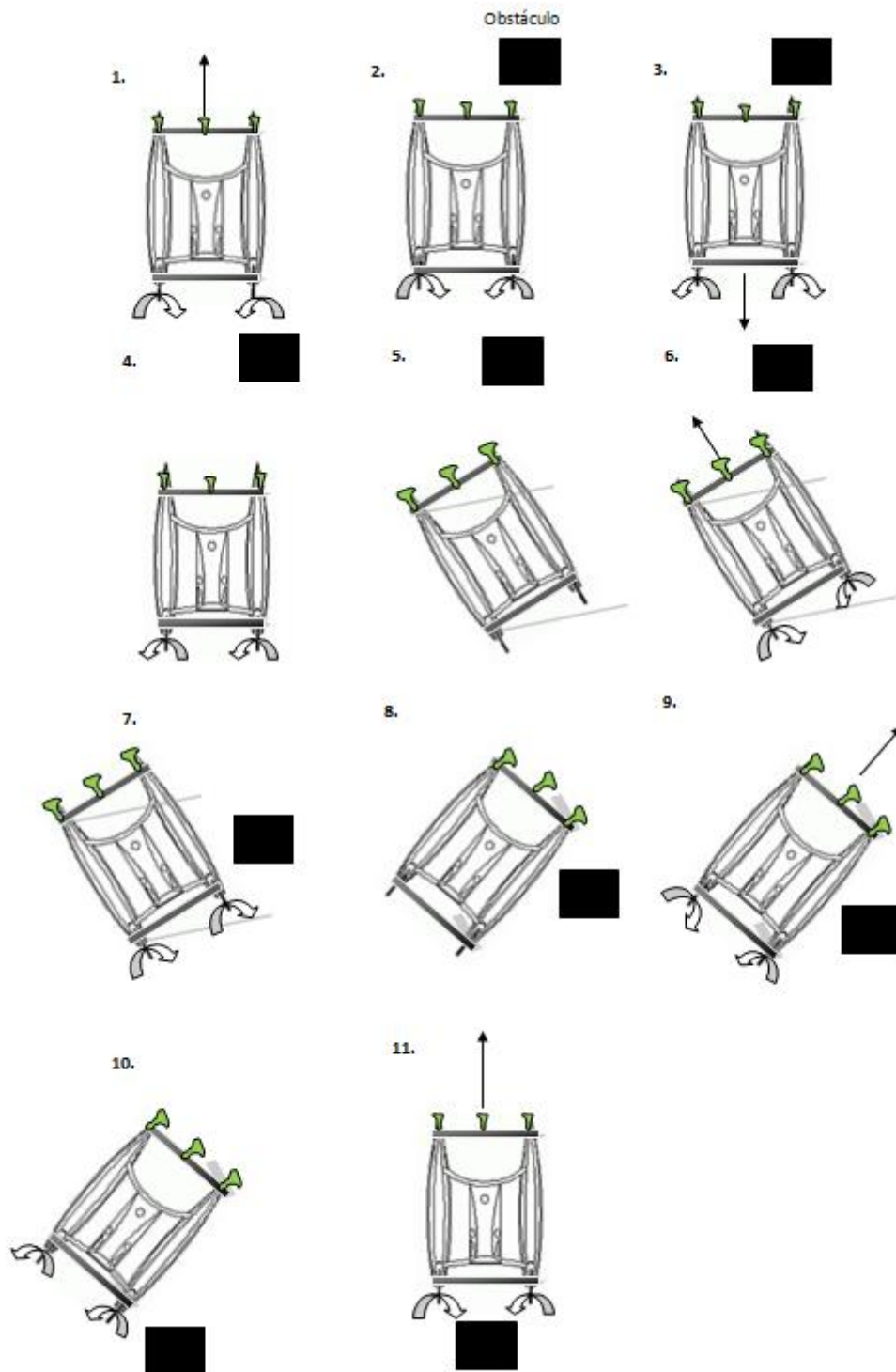


Figura 47. Acciones del buque durante el esquivado de obstáculos

El código utilizado durante la prueba se encuentra en el Anexo A.6. Código para evitar colisiones con obstáculos, es muy similar al anteriormente presentado pero varía la definición de las variables y el bucle loop.

La programación del código es mucho más compleja que los anteriores, se basa en la comparación de las distancias obtenidas en los sensores y actuar según ellas. Para realizar la prueba con éxito se añadieron otras variables como turnRight2 y turnLeft2.

Hay que recalcar, que se realizaron vídeos, que corroboran el algoritmo de esquivar obstáculos, pero sin llegar a lograrlo, ya que el viento que hubo durante la realización de las pruebas era muy elevado, además de la gran superficie del catamarán que aumentaba el efecto negativo del viento.

## 6.10 Dificultades

Durante las pruebas de comportamiento de la embarcación, hubieron algunos contratiempos que sucedieron de forma imprevista:

- El buque era susceptible de modificar su dirección a causa del viento y la corriente.
- El comportamiento de la nave variaba según la energía aportada por la fuente de alimentación.
- La capacidad de giro con dos motores no era tan elevada como se planteaba.
- La efectividad de los sensores de ultrasonidos es alta, pero no lo suficiente como para evitar la percepción de falsas referencias, causando posibles accidentes.

Hubieron dificultades como la baja disponibilidad de instalaciones donde poder hacer las pruebas, que en mi caso, al no tener piscina, tuve que pedir ayuda a conocidos, entidades públicas y privadas. Las personas que por casualidad tenían una piscina, esta era comunitaria. Al realizarse en primavera en TFG, muchas de ellas estaban en mantenimiento o habían vecinos utilizando las instalaciones, por lo que no se podía efectuar ninguna prueba.

Cuando se pedía ayuda a entidades públicas como la piscina municipal de Badalona, habían que hacer trámites que duraban meses y finalmente no eran respondidos. Incluso, en lugares públicos como lo son el parque de Can Zam de Santa Coloma, no estaba permitido por la introducción de una nueva fauna. Gracias a entidades privadas como el Club Náutico de Badalona y personas cercanas pude realizar las pruebas.



Finalmente, durante la realización de la prueba en la que la nave debería ser capaz de evitar obstáculos, las rachas de viento no permitieron lograrla con éxito, a pesar de que las acciones del buque fueron correctas.

### **6.11 Alternativas**

Durante el desarrollo de circuitos y códigos hubieron algunos que no se llevaron a cabo porque no llegaron a funcionar. Como por ejemplo, el circuito de control variable del motor DC a través de un transistor

La idea inicial de este circuito era poder variar las revoluciones por minuto de un motor DC, para ello se ha utilizado:

- 1 transistor NPN 2N2222A capaz de soportar 50 Voltios
- 1 motor de corriente continua de 9 V
- 1 pila de 9 V, que administraría electricidad al motor
- 1 resistencia de 270 Ohms.

No obstante, a la hora de realizar el circuito, la pila producía un pitido a velocidades bajas y no permitía que el motor funcionase. En cambio, para velocidades altas, se creaba un arco desde la pila al cable conductor de forma que funcionaba el motor pero descontroladamente.

El circuito realizado es el siguiente:

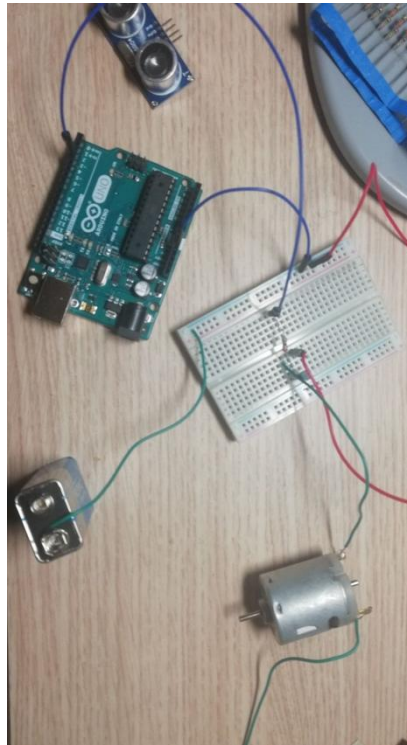


Figura 48. Circuito de control variable del motor DC

## Capítulo 7: Presupuesto

El presupuesto utilizado durante el proyecto, ha sido supervisado por la profesora Rosa M. Fernández durante todo el proyecto. El material comprado se ha dividido en tres partes, la impresión de los módulos que forman la estructura externa, elementos externos y el material electrónico y funcional.

### 7.1 Módulos

Los módulos son piezas que se deben unir para formar la pieza externa del buque. En la impresión de los 12 módulos se han invertido 216 horas durante dos semanas, de las cuales 54 horas se emplean en las piezas más grandes del casco. Inicialmente, estaba pensado llevarlas a cabo en 3 semanas, pero se redujo el tiempo al dejar las impresoras 3D encendidas durante la noche.

La cantidad total de filamento necesaria para completar el casco son 800 gramos, con un precio de 69,34€ solamente en material.

DESCRIPCIÓN	CANTIDAD	PRECIO UNIDAD	SUBTOTAL	DTO.	TOTAL
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	9,7000	3,7200	36,08		36,08
FILAMENT PLA (gr)	71,0000	0,0450	3,20		3,20
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	17,0000	3,7200	63,24		63,24
FILAMENT PLA (gr)	121,0000	0,0450	5,45		5,45
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	27,0000	3,7200	100,44		100,44
FILAMENT PLA (gr)	192,0000	0,0450	8,64		8,64
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	24,0000	3,7200	89,28		89,28
FILAMENT PLA (gr)	166,0000	0,0450	7,47		7,47
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	18,0000	3,7200	66,96		66,96
FILAMENT PLA (gr)	127,0000	0,0450	5,72		5,72
TEMPS D'IMPRESSIÓ (TARIFA EDU) (h)	12,5000	3,7200	46,50		46,50
FILAMENT PLA (gr)	93,0000	0,0450	4,19		4,19
TÈCNIC MANIP.	4,0000	9,0900	36,36		36,36

BASE	I.V.A.
473,53	99,44

<b>TOTAL:</b>	<b>572,97</b>
---------------	---------------

Tabla 9. Precio total de la impresión de los módulos

Debido a que no poseía ninguna impresora 3D, se ha solicitado un técnico que manipule las piezas durante todo el procedimiento. El precio total de toda la impresión ha sido de 572,97€ y ha sido efectuada en BDN Labs, una empresa de creación de proyectos mediante impresoras 3D.

## 7.2 Elementos de externos

Los elementos que se han utilizado durante la unión de todas las piezas del barco se puede dividir en 2 partes: preparación del casco a través de pegamentos y sustancias, y los elementos sujeción

### 7.2.1 Preparación del casco a través de pegamentos y sustancias

Durante la unión del casco, han sido necesarios ciertos componentes que aporten la resistencia estructural necesaria, además de elementos para aplicarlos. Estos han sido:

- Pegamento Bicomponente
- Masilla de poliéster
- Imprimación
- Pintura
- Papel de lija al agua grado 500
- Papel de lija al agua grado 320
- Papel de lija al agua grado 240
- Papel de lija al agua grado 100
- Espátulas de carroceros inoxidables

Material	Precio (€)
Pegamento Bicomponente	30.66
Masilla de poliéster	16.16
Imprimación	10.15
Pintura	14.34
Papel de lija al agua grado 500	0.62
Papel de lija al agua grado 320	0.58
Papel de lija al agua grado 240	0.74

Papel de lija al agua grado 100	0.62
Espátulas de carroceros inoxidables	4.71

Tabla 10. Material de unión del casco

El precio total de estos productos ha sido de: 78.58€

### 7.2.2 Elementos de sujeción

Debido a que se han utilizado dispositivos electrónicos que hacen funcionar el buque como autónomo, es necesario protegerlos de los ambientes húmedos como el agua.

Primero, se han reutilizado unos tabloncillos de madera, que se han tallado a medida para unir cada uno de las palas. Después se ha creado una caja de PVC que protege el material electrónico sostenida sobre una barra de aluminio. La caja se ha formado a partir de un perfil en L y una placa de PVC. Las dimensiones se encuentran en la Figura 22.

Cantidad	Material	Precio (€)
1	Barra en perfil L de PVC	2.99
1	Sierra	4.95
2	Hojas de la sierra	1.75
1	Placa de PVC	8.95
1	Barra de aluminio	8.00

Tabla 11. Material de sujeción

El precio total de estos productos ha sido de: 26.64€

Finalmente, debido a que los motores necesitaban una sujeción para transmitir de forma óptima el empuje, se han creado dos arbotantes. El material ha sido reutilizado de unos listones y dos cuñas de madera.

## 7.3 Material electrónico y funcional

A diferencia de los casos anteriores, el material electrónico es esencial y se puede reducir solamente el presupuesto escogiendo opciones más asequibles.

Todo el material electrónico utilizado ha sido el siguiente:

Cantidad	Material	Precio (€)
3	Sensores de ultrasonidos	7.44
1	Cables Macho-Hembra	3.30
1	Cables Macho-Macho	3.30
1	Cables Hembra-Hembra	3.30
1	Paquete Arduino	73.55
1	Doble puente H	8.27
2	Cardans	48.00
2	Ejes de transmisión	25.00
1	Hélice dextrógira	5.00
1	Hélice levógira	5.00
9	Pilas de alimentación de 9V	27.00

Tabla 12. Material electrónico

El precio de todos los elementos ha sido de: 209.16€

Finalmente, el precio total teniendo en cuenta todas las etapas del proyectos, ha sido de: 887,35€. El resultado hubiese variado modificando las dimensiones de la embarcación, ya que la impresión 3D ha sido el principal consumidor de los recursos disponibles. Por otra parte, hay elementos como los cardans o ejes de transmisión que no permite una alternativa más asequible debido a que solamente hay una tienda en Barcelona que pueda proveerlos.

## Capítulo 8: Conclusiones y líneas futuras de investigación

La evolución de los buques autónomos está siendo liderada por empresas como Rolls Royce que pretenden cambiar en los próximos años todo el diseño de embarcaciones para optimizarlas y se modifiquen los puertos para adaptarse a ellos, tanto en los puestos de trabajo como en las infraestructuras. Organizaciones como la OMI ya están desarrollando directrices para la creación de buques autónomos y operaciones sin tripulación como muestra de cambio en un futuro no muy lejano.

Durante el planteamiento del prototipo desarrollado en el presente trabajo se establecieron unos objetivos que se debían alcanzar, el resultado ha sido el siguiente:

El diseño del buque ha sido realizado con éxito, durante las pruebas no ha habido ningún problema relacionado con la flotabilidad ni la estabilidad.

Durante el ensamblaje de los módulos, a pesar de que el casco ha sufrido una pequeña modificación en la superficie al haberlo dejado en el exterior cuando recientemente se había pintado, no ha tenido ningún tipo de consecuencia en los resultados de las pruebas ni en la resistencia estructural.

El diseño e implementación del circuito Arduino se ha efectuado con algunos contratiempos, como por ejemplo hallar elementos como los cardans o los ejes de transmisión, que fueron resueltos satisfactoriamente. Sin embargo, el algoritmo de control y navegación con la plataforma Arduino se realizó con éxito.

El código que se ha programado para realizar las pruebas de navegación ha sido creado sin ningún tipo de conocimiento previo. A partir de libros e información de internet, se ha logrado escribir el código encargado de realizar todas las acciones para que el buque sea autónomo.

Se ha logrado conocer el comportamiento del buque en todo tipo de situaciones en aguas tranquilas, ya sea el avance, giro o freno de la embarcación.

Se ha alcanzado la meta de conseguir que el buque se mantenga en la piscina sin colisionar con los bordes de esta, y que pudiese ir a un punto de la piscina y volver sin colisionar una vez más con los bordes de la piscina.

Después de cada una de las etapas de estudio y desarrollo, en las que se ha seleccionado elementos de adherencia para los módulos e instrumentación y se ha desarrollado un código

específico para este proyecto, finalmente se ha logrado con éxito la construcción desde cero de un vehículo acuático autónomo, equipado electrónicamente con la capacidad de evitar obstáculos.

Los resultados obtenidos han sido excelentes desde el punto de vista de la flotabilidad, estabilidad, resistencia estructural y tiempo de reacción de los sensores y motores. Como líneas futuras de investigación, este proyecto puede servir para otros estudiantes tanto de grado como de máster, para realizar trabajos más complejos añadiendo otros dispositivos y alcanzando objetivos más lejanos, como puede ser añadirle un GPS (*Global Position System*) y que realice una ruta exacta evadiendo todos los obstáculos.

Es más, incluso puede servir para realizar prácticas para los estudiantes de la Facultad Náutica de Barcelona, para analizar el comportamiento hidrodinámico de las embarcaciones o para aprender a realizar códigos con Arduino y relacionarse con los dispositivos que posee.



## Bibliografía y referencias

[1] Embarcaciones de recreo autónomas. 24/02/2017

Disponible en: <http://innovacion.portsdebalears.com/idea-innovadora/embarcaciones-de-recreo-autonomas/>

[2] ¿Conoces los barcos autónomos? 29/09/2016

Disponible en: <https://tecvolucion.com/conoces-los-barcos-autonomos/>

[3] Los barcos autónomos amenazan el trabajo de la tripulación de los buques de carga. 19/06/2017.

Disponible en: <https://www.technologyreview.es/s/7947/los-barcos-autonomos-amenazan-el-trabajo-de-la-tripulacion-de-los-buques-de-carga>

[4] Rolls-Royce plantea un futuro sin marineros. 22/06/2016.

Disponible en: <https://www.xataka.com/vehiculos/rolls-royce-plantea-un-futuro-sin-marineros-embarcaciones-autonomas-y-un-potente-centro-de-control-en-tierra>

[5] El futuro del transporte marítimo. 26/05/2017.

Disponible en: <https://www.infobae.com/america/wapo/2017/05/26/el-futuro-del-transporte-maritimo-como-funcionaran-los-nuevos-barcos-sin-tripulacion/>

[6] Barcos autónomos: la robótica a la conquista de los mares. 6/06/2018.

Disponible en: <https://www.imnovation-hub.com/es/transformacion-digital/barcos-autonomos-conquista-de-los-mares/>

[7] Detector de obstáculos con sensor de infrarrojo y Arduino. 2/06/2016.

Disponible en: <https://www.luisllamas.es/detectar-obstaculos-con-sensor-infrarrojo-y-arduino/>

[8] Medir distancia con Arduino y sensor de ultrasonidos HC-SR04. 16/06/2015.

Disponible en: <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>

[9] Cuatro alternativas a Arduino. 9/01/2013.

Disponible en: <https://blogthinkbig.com/4-alternativas-arduino-beaglebone-raspberrypi-nanode-waspmote>

[10] ¿Qué es Arduino?

Disponible en: <https://descubrearduino.com/>

[11] OMI inicia la validación de operaciones futuras en naves autónomas. 21/05/2018

Disponible en: <https://portalportuario.cl/omi-inicia-trabajo-para-validar-operacion-de-futuras-naves-autonomas/>

[12] Código de prácticas para buques autónomos. 20/11/2017

Disponible en: <https://sectormaritimo.es/codigo-practicas-buques-autonomos>

[13] El primer buque de contenedores eléctrico. 7/11/2017

Disponible en: <https://newatlas.com/autonomous-electric-shipping-container-vessel/49477/>

[14] Barcos sin capitán. 4/10/2017

Disponible en: [https://as.com/betech/2017/10/04/portada/1507113715\\_389810.html](https://as.com/betech/2017/10/04/portada/1507113715_389810.html)

[15] Proyecto de Yara 14/8/2017

Disponible en: [https://www.yara.com/news-and-media/press\\_releases/2103105/press\\_release/201705/yara-and-kongsberg-enter-into-partnership-to-build-worlds-first-autonomous-and-zero-emissions-ship/](https://www.yara.com/news-and-media/press_releases/2103105/press_release/201705/yara-and-kongsberg-enter-into-partnership-to-build-worlds-first-autonomous-and-zero-emissions-ship/)

[16] Proyecto de Rolls Royce 12/9/2017

Disponible en: <https://www.rolls-royce.com/media/our-stories/press-releases/2017/12-09-2017-rr-reveals-plans-for-autonomous-naval-vessel.aspx>

[17] Rolls-Royce y Google 04/10/2017

Disponible en: <http://www.ciospain.es/industria-y-utilities/rollsroyce-y-google-cloud-trabajan-para-hacer-realidad-la-navegacion-autonoma>

[18] Normas del transporte marítimo.

Disponible en: <http://www.europarl.europa.eu/factsheets/es/sheet/125/el-transporte-maritimo-normas-de-trafico-y-de-seguridad>

[19] Comité de seguridad marítima (MSC), 99º periodo de sesiones

Disponible en:

<http://www.imo.org/es/MediaCentre/IMOMediaAccreditation/Paginas/MSC99preview.aspx>

[20] Catamaranes

Disponible en: <https://www.topbarcos.com/barcos-ocasion-brokerage?tipo=catamaran>

[21] Scott, F., (2014), El libro de proyectos de Arduino. Arduino Starter Kit, Italia.

[22] Daniel, L., (2017), Arduino Práctico. Anaya Multimedia.

[23] Jeremy, B., (2014), Arduino a fondo. Anaya Multimedia.

[24] Raspberry Pi y sus límites. 21/10/2012

Disponible en: <https://www.xataka.com/componentes/con-la-raspberry-pi-el-limite-lo-pones-tu-aprende-que-puedes-hacer-ii>

[25] La inteligencia de los buques. 20/03/2018

Disponible en: <https://www.rolls-royce.com/products-and-services/marine/ship-intelligence.aspx#section-contacts>

## ANEXOS

### A.1 Alarma mediante ultrasonidos a cierta distancia

Para este circuito se necesitan muy pocos componentes:

- 1 sensor de ultrasonidos
- 1 LED

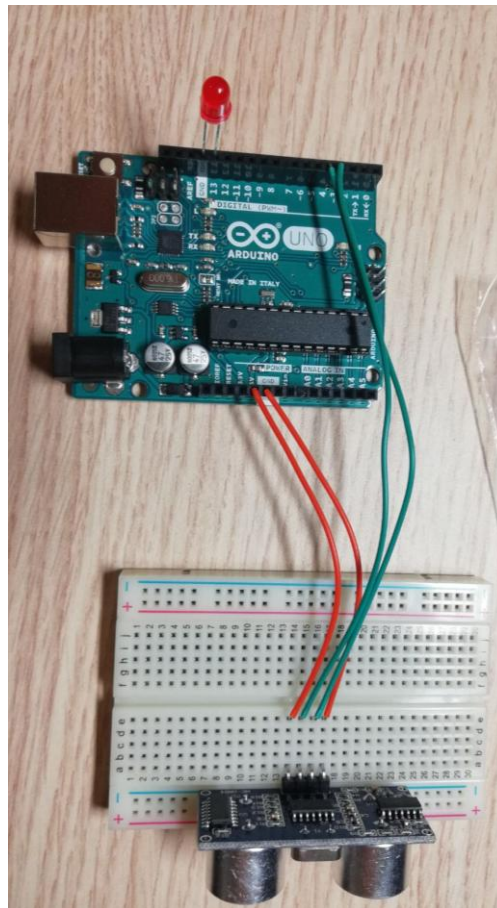


Figura 49. Alarma mediante ultrasonidos a cierta distancia

El sensor de ultrasonidos es un dispositivo prefabricado por Arduino que, mediante el siguiente código, es capaz de encender el LED según la distancia a la que se localicen obstáculos.

```

sketch_feb22a
#define Pecho 4
#define Ptrig 3
long duracion, distancia;

void setup() {
  Serial.begin (9600);
  pinMode(Pecho, INPUT);
  pinMode(Ptrig, OUTPUT);
  pinMode(13,1);
}

void loop() {
  digitalWrite(Ptrig, LOW);
  delayMicroseconds(2);
  digitalWrite(Ptrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Ptrig, LOW);

  duracion = pulseIn(Pecho, HIGH);
  distancia = (duracion/2) / 29;

  if (distancia >= 500 || distancia <= 0){
    Serial.println("---");
  }
  else {
    Serial.print(distancia);
    Serial.println("cm");
    digitalWrite(13,0);
  }

  if (distancia <= 10 && distancia >= 1){
    digitalWrite(13, 1);
    Serial.println("Alarma.....");
  }
  delay(400);
}

```

Figura 50. Código de ultrasonidos

Hay que tener en cuenta que el voltaje de alimentación es de 5V, con un rango de medición que oscila desde los 2 cm a 5 metros, que la distancia de referencia se debe dividir entre dos, ya que la onda recorre dos veces el mismo trayecto, y que la velocidad del sonido es 1/29 cm/us, por lo que la distancia debe dividirse entre la velocidad del sonido.

$$Distancia = \frac{\left(\frac{Duración}{2}\right)}{29} \quad [5]$$

Distancia = Longitud recorrida por la onda.

Duración = Tiempo que tarda la ola en recorrer una distancia.

Cada pulso tarda 10 microsegundos en realizarse, es un dato a tener en cuenta a la hora de programar.

## A.2 Modificación de la velocidad de los motores con PWM y doble puente H

Para este proyecto se utilizan dos motores para poder modificar la dirección del buque dependiendo solo de la potencia de los motores. Las RPMs de los motores DC variarán debido a los PWM y al doble puente H. La técnica PWM (*pulse-width modulation*) permite modular el ancho de los pulsos de una señal, las conexiones PWM se encuentran en la placa Arduino en los números con el símbolo (~).

Además, al encontrarse con un objeto próximo a la embarcación, el sensor de ultrasonidos tiene la capacidad de moverse 180 grados desde su posición inicial, por lo que puede detectar los objetos a su alrededor, en caso de que haya localizado algún obstáculo, el ultrasonidos enviará la información al Arduino para que modifique la velocidad de los motores.

### Componentes:

- Doble puente H (L298N)
- Arduino UNO
- Servomotor
- Sensor de ultrasonidos (HC-SR04)
- Dos pilas de 9V
- Cables
- Motores DC 9V
- Protoboard

### Conexiones:

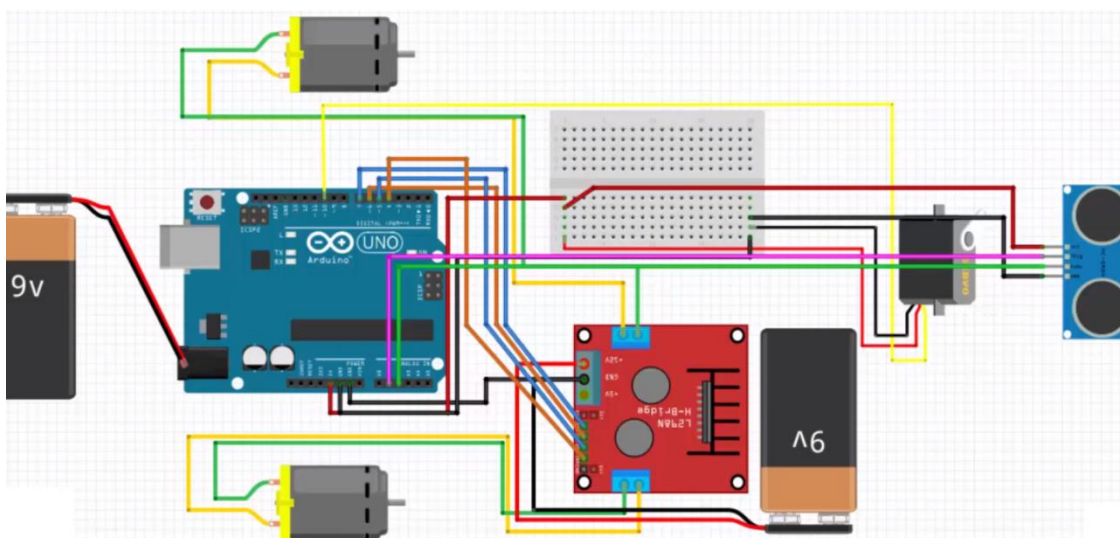
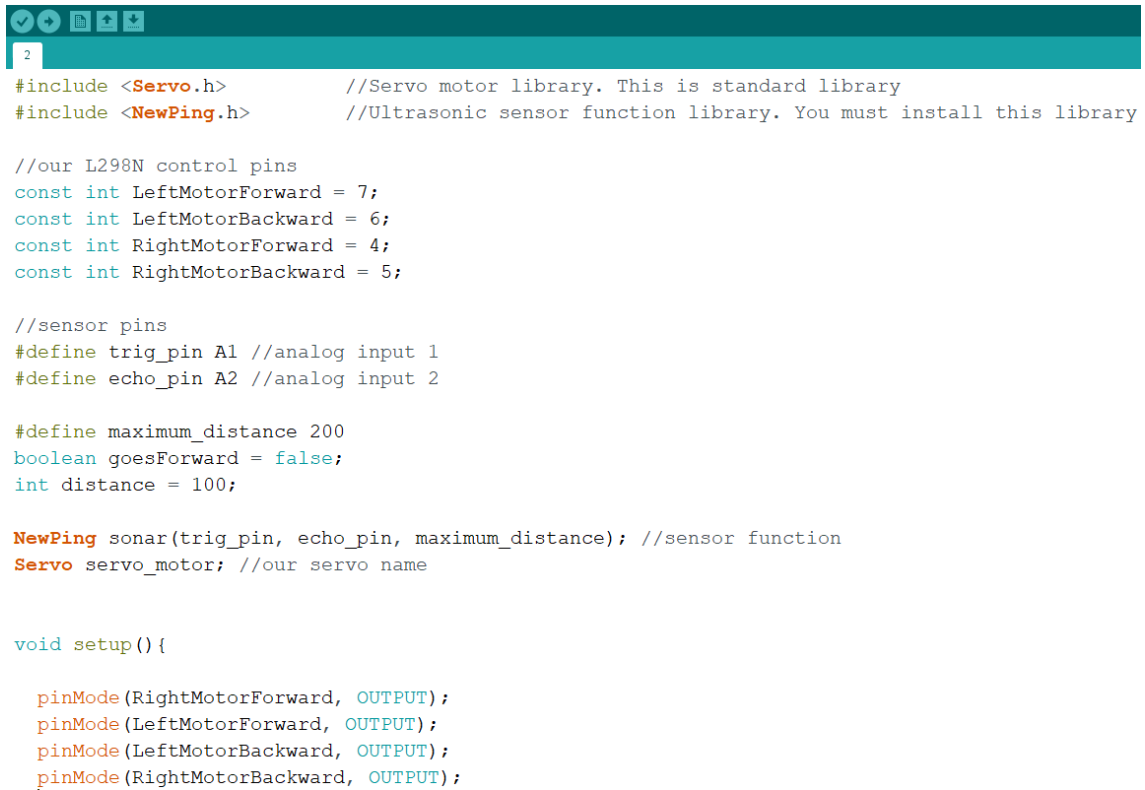


Figura 51. Conexión para motores mediante PWM

## Código:



```
#include <Servo.h>           //Servo motor library. This is standard library
#include <NewPing.h>         //Ultrasonic sensor function library. You must install this library

//our L298N control pins
const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 4;
const int RightMotorBackward = 5;

//sensor pins
#define trig_pin A1 //analog input 1
#define echo_pin A2 //analog input 2

#define maximum_distance 200
boolean goesForward = false;
int distance = 100;

NewPing sonar(trig_pin, echo_pin, maximum_distance); //sensor function
Servo servo_motor; //our servo name

void setup(){

  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);
```

```
servo_motor.attach(10); //our servo pin

servo_motor.write(115);
delay(2000);
distance = readPing();
delay(100);
distance = readPing();
delay(100);
distance = readPing();
delay(100);
distance = readPing();
delay(100);
}

void loop(){

  int distanceRight = 0;
  int distanceLeft = 0;
  delay(50);

  if (distance <= 20){
    moveStop();
    delay(300);
    moveBackward();
    delay(400);
    moveStop();
    delay(300);
  }
}
```



```

        distanceRight = lookRight();
        delay(300);
        distanceLeft = lookLeft();
        delay(300);

        if (distance >= distanceLeft){
            turnRight();
            moveStop();
        }
        else{
            turnLeft();
            moveStop();
        }
    }
    else{
        moveForward();
    }
    distance = readPing();
}

int lookRight(){
    servo_motor.write(50);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
}

int lookLeft(){
    servo_motor.write(170);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
    delay(100);
}

int readPing(){
    delay(70);
    int cm = sonar.ping_cm();
    if (cm==0){
        cm=250;
    }
    return cm;
}

void moveStop(){

    digitalWrite(RightMotorForward, LOW);
    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);
    digitalWrite(LeftMotorBackward, LOW);
}

void moveForward(){

```

```
if(!goesForward){

    goesForward=true;

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
}
}

void moveBackward(){

    goesForward=false;

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorForward, LOW);

}

void turnRight(){

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorForward, LOW);

    delay(500);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);

}

void turnLeft(){

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);

    delay(500);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);

}
```

Figura 52. Código para motores mediante PWM

**Explicación:**

El código anterior permite al sensor de ultrasonidos ir detectando los objetos que se interpongan en la dirección del catamarán. Una vez detectados, los motores variarán su velocidad e incluso cambiarán el sentido de giro, para evitar el choque con cualquier objeto que se encuentre en la trayectoria.

Todo esto se consigue con la función loop, que permite estar enviando constantemente ondas para estar alerta por un posible impacto.

### A.3 Giro con los dos motores

El código de la prueba de giro con los dos motores es muy sencillo, simplemente se ordena que a una distancia mayor de 1 metro de cualquier obstáculo, uno de los motores se invierta para poder realizar la maniobra. Con la intención de que la prueba no sea errónea por cualquiera falsa referencia, se programó de tal manera que la onda del sensor de ultrasonidos siempre diese la mayor distancia posible, 5 metros. Para ello, a la hora de elegir las salidas y entradas de los sensores, se eligieron valores incorrectos, se puede observar en el inicio del código (#define trig\_pin/echo\_pin).

```
#include <Servo.h>           //Servo motor library. This is standard library
#include <NewPing.h>         //Ultrasonic sensor function library. You must install this library

//our L298N control pins
const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 4;
const int RightMotorBackward = 5;

//sensor pins
#define trig_pin A1 //analog input 1
#define echo_pin A2 //analog input 2

#define maximum_distance 350 // 200
boolean goesForward = false;
int distance = 200; //100

NewPing sonar(trig_pin, echo_pin, maximum_distance); //sensor function
Servo servo_motor; //our servo name

void setup(){

    Serial.begin(9600);
    pinMode(RightMotorForward, OUTPUT);
    pinMode(LeftMotorForward, OUTPUT);
    pinMode(LeftMotorBackward, OUTPUT);
    pinMode(RightMotorBackward, OUTPUT);

    servo_motor.attach(10); //our servo pin

    servo_motor.write(115);
    delay(2000);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
}
```

```
void loop(){

    int distanceRight = 0;
    int distanceLeft = 0;
    delay(50);

    if (distance >= 100){
        turnLeft();
    }
    else{
        goesForward=false;
        moveForward();
    }
    distance = readPing();
    Serial.println(distance);
}

int lookRight(){
    servo_motor.write(50);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
}

int lookLeft(){
    servo_motor.write(170);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
    delay(100);
}

int readPing(){
    delay(70);
    int cm = sonar.ping_cm();
    if (cm==0){
        cm=250;
    }
}
```

```

    }
    return cm;
}

void moveStop() {

    digitalWrite(RightMotorForward, LOW);
    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);
    digitalWrite(LeftMotorBackward, LOW);
}

void moveForward() {

    if(!goesForward) {

        goesForward=true;

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorBackward, LOW);
    }
}

void moveBackward() {

    goesForward=false;

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorForward, LOW);

}

void turnRight() {

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorForward, LOW);

    delay(4000);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);

}

void turnLeft() {

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);

    delay(20000);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
}

```

Figura 53. Código de giro con los dos motores

## A.4 Freno con los dos motores invertidos

La dificultad de la programación del código de la prueba de freno con los dos motores, es escasa, debido a que la actividad a realizar es simple. Todas las acciones que se habían escrito en el anterior código, se mantienen, solamente la información que contiene el loop. En este caso, al querer analizar la desaceleración del buque en frente a un obstáculo, simplemente se aplica la acción de inversión de motores "moveBackward" que se activa a la distancia que nos convenga.

```
#include <Servo.h>           //Servo motor library. This is standard library
#include <NewPing.h>         //Ultrasonic sensor function library. You must install this library

//our L298N control pins
const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 4;
const int RightMotorBackward = 5;

//sensor pins
#define trig_pin A2 //analog input 1
#define echo_pin A3 //analog input 2

#define maximum_distance 200 // 200
boolean goesForward = false;
int distance = 100; //100

NewPing sonar(trig_pin, echo_pin, maximum_distance); //sensor function
Servo servo_motor; //our servo name

void setup(){

    Serial.begin(9600);
    pinMode(RightMotorForward, OUTPUT);
    pinMode(LeftMotorForward, OUTPUT);
    pinMode(LeftMotorBackward, OUTPUT);
    pinMode(RightMotorBackward, OUTPUT);

    servo_motor.attach(10); //our servo pin

    servo_motor.write(115);
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
}
```

```
void loop() {

    int distanceRight = 0;
    int distanceLeft = 0;
    delay(50);

    if (distance <= 100) {
        moveBackward();
    }
    else{
        goesForward=false;
        moveForward();
    }
    distance = readPing();
    Serial.println(distance);
}

int lookRight(){
    servo_motor.write(50);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
}

int lookLeft(){
    servo_motor.write(170);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
    delay(100);
}

int readPing(){
    delay(70);
    int cm = sonar.ping_cm();
    if (cm==0){
        cm=200;
    }
}
```



```

    }
    return cm;
}

void moveStop() {

    digitalWrite(RightMotorForward, LOW);
    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);
    digitalWrite(LeftMotorBackward, LOW);
}

void moveForward() {

    if(!goesForward){

        goesForward=true;

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorBackward, LOW);
    }
}

void moveBackward() {

    goesForward=false;

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorForward, LOW);
}

void turnRight() {

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorForward, LOW);

    delay(4000);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);

}

void turnLeft() {

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);

    delay(4000);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
}

```

Figura 54. Código de freno con los dos motores



## A.5 Código para evitar la colisión con el borde de la piscina

Después de haber realizado las pruebas de giro, de avance y de freno, se obtuvieron los valores que permitían conocer los límites de la embarcación, y utilizarla en el código para que la prueba resultase exitosa. A diferencia de los otros códigos, este precisaba de más complejidad, ya que habían que sincronizar 3 sensores de ultrasonidos simultáneamente. Simplemente se programó, según el diagrama de estados que se encuentra en la Figura 43. Los resultados inicialmente fueron confusos, ya que los motores giraban en sentido contrario al escrito en el código, después de cierto tiempo, se decidió invertir las acciones de avance y inversión de motores, para que concordase con la intención de la prueba.

```
#include <Servo.h>           //Servo motor library. This is standard library
#include <NewPing.h>         //Ultrasonic sensor function library. You must install this library

//our L298N control pins
const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 4;
const int RightMotorBackward = 5;

//sensor pins
#define trig_pin1 A0 //analog input 1  Sensor Babor
#define echo_pin1 A1 //analog input 2  Sensor Babor
#define trig_pin2 A2 //analog input 3  Sensor Central
#define echo_pin2 A3 //analog input 4  Sensor Central
#define trig_pin3 A4 //analog input 5  Sensor Estribor
#define echo_pin3 A5 //analog input 6  Sensor Estribor

#define maximum_distance 500    // 200
boolean goesForward = false;
int distance1 = 500;    //100
int distance2 = 500;
int distance3 = 500;

NewPing sonar1(trig_pin1, echo_pin1, maximum_distance); //Sensor Function Babor
NewPing sonar2(trig_pin2, echo_pin2, maximum_distance); //Sensor Function Central
NewPing sonar3(trig_pin3, echo_pin3, maximum_distance); //Sensor Function Estribor
Servo servo_motor; //our servo name

void setup() {

  Serial.begin(9600);

  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);

  servo_motor.attach(10); //our servo pin

  servo_motor.write(115);
  delay(2000);
```

```
distance1 = readPing1();
delay(100);
distance2 = readPing2();
delay(100);
distance3 = readPing3();
delay(100);

}

void loop() {

  int distanceRight = 0;
  int distanceLeft = 0;
  delay(50);

  Serial.print("d1=");
  Serial.println(distance1);

  Serial.print("d2=");
  Serial.println(distance2);

  Serial.print("d3=");
  Serial.println(distance3);

  if (distance1 <= 400) {
    moveBackward();
    delay(10000);
    if (distance3 >= distance1) {
      turnRight();
      moveStop();
    }
    else {
      turnLeft();
      moveStop();
    }
  }
  else {
    moveForward();
  }
}
```

```
if (distance2 <= 400){
    moveBackward();
    delay(10000);
    if (distance3 >= distance1){
        turnRight();
        moveStop();
    }
    else{
        turnLeft();
        moveStop();
    }
}
else{
    moveForward();
}
if (distance3 <= 400){
    moveBackward();
    delay(10000);
    if (distance1 >= distance3){
        turnLeft();
        moveStop();
    }
    else{
        turnRight();
        moveStop();
    }
}
else{
    moveForward();
}
distance1 = readPing1();
distance2 = readPing2();
distance3 = readPing3();
}

int readPing1(){
    delay(70);
    int cm = sonar1.ping_cm();
    if (cm==0){
```

```
        cm=500;
    }
    return cm;
}

int readPing2(){
    delay(70);
    int cm = sonar2.ping_cm();
    if (cm==0){
        cm=500;
    }
    return cm;
}

int readPing3(){
    delay(70);
    int cm = sonar3.ping_cm();
    if (cm==0){
        cm=500;
    }
    return cm;
}

void moveStop(){

    digitalWrite(RightMotorForward, LOW);
    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);
    digitalWrite(LeftMotorBackward, LOW);
}

void moveForward(){

    if(!goesForward){

        goesForward=true;

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorBackward, LOW);
    }
}
```

```
    }

    void moveBackward() {

        goesForward=false;

        digitalWrite(LeftMotorBackward, HIGH);
        digitalWrite(RightMotorBackward, HIGH);

        digitalWrite(LeftMotorForward, LOW);
        digitalWrite(RightMotorForward, LOW);

    }

    void turnRight() {

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorBackward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorForward, LOW);

        delay(9000);

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorBackward, LOW);

    }

    void turnLeft() {

        digitalWrite(LeftMotorBackward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorForward, LOW);
        digitalWrite(RightMotorBackward, LOW);

        delay(9000);

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorBackward, LOW);

    }
}
```

Figura 55. Código para evitar la colisión con el borde

## A.6 Código para evitar colisiones y seguir la ruta

El código se basa en el cálculo de tiempos que tarda la embarcación en girar y en frenar, teniendo en cuenta los tiempos estimado de cada acción, era posible evitar un obstáculo por tiempos, eso sí, teniendo presente que el obstáculo estuviese sin moverse.

Según avancen las pruebas, se irían ajustando los tiempos de reacción para lograr que evitase la colisión óptimamente.

```
#include <Servo.h>           //Servo motor library. This is standard library
#include <NewPing.h>         //Ultrasonic sensor function library. You must install this library

//our L298N control pins
const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 4;
const int RightMotorBackward = 5;

//sensor pins
#define trig_pin1 A0 //analog input 1  Sensor Babor
#define echo_pin1 A1 //analog input 2  Sensor Babor
#define trig_pin2 A2 //analog input 3  Sensor Central
#define echo_pin2 A3 //analog input 4  Sensor Central
#define trig_pin3 A4 //analog input 5  Sensor Estribor
#define echo_pin3 A5 //analog input 6  Sensor Estribor

#define maximum_distance 500      // 200
boolean goesForward = false;
int distance1 = 500;    //100
int distance2 = 500;
int distance3 = 500;

NewPing sonar1(trig_pin1, echo_pin1, maximum_distance); //Sensor Function Babor
NewPing sonar2(trig_pin2, echo_pin2, maximum_distance); //Sensor Function Central
NewPing sonar3(trig_pin3, echo_pin3, maximum_distance); //Sensor Function Estribor
Servo servo_motor; //our servo name

void setup() {

  Serial.begin(9600);

  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);

  servo_motor.attach(10); //our servo pin

  servo_motor.write(115);
  delay(2000);
```



```
distance1 = readPing1();
delay(100);
distance2 = readPing2();
delay(100);
distance3 = readPing3();
delay(100);

}

void loop(){

  int distanceRight = 0;
  int distanceLeft = 0;
  delay(50);

  Serial.print("d1=");
  Serial.println(distance1);

  Serial.print("d2=");
  Serial.println(distance2);

  Serial.print("d3=");
  Serial.println(distance3);

  if (distance1 <= 400){
    moveBackward();
    delay(10000);
    if (distance3 >= distance1){
      turnRight();
      moveBackward();
      turnLeft2();
      moveBackward();
      turnRight();
    }
    else{
      turnLeft();
      moveBackward();
      turnRight2();
      moveBackward();
    }
  }
}
```

```
        turnLeft();
    }
}
else{
    moveForward();
}
if (distance2 <= 400){
    moveBackward();
    delay(10000);
    if (distance3 >= distance1){
        turnRight();
        moveBackward();
        turnLeft2();
        moveBackward();
        turnRight();
    }
    else{
        turnLeft();
        moveBackward();
        turnRight2();
        moveBackward();
        turnLeft();
    }
}
else{
    moveForward();
}
if (distance3 <= 400){
    moveBackward();
    delay(10000);
    if (distance1 >= distance3){
        turnLeft();
        moveBackward();
        turnRight2();
        moveBackward();
        turnLeft();
    }
    else{
        turnRight();
    }
}
```

```
        moveBackward();
        turnLeft2();
        moveBackward();
        turnRight();
    }
}
else{
    moveForward();
}
distance1 = readPing1();
distance2 = readPing2();
distance3 = readPing3();
}

int readPing1(){
    delay(70);
    int cm = sonar1.ping_cm();
    if (cm==0){
        cm=500;
    }
    return cm;
}

int readPing2(){
    delay(70);
    int cm = sonar2.ping_cm();
    if (cm==0){
        cm=500;
    }
    return cm;
}

int readPing3(){
    delay(70);
    int cm = sonar3.ping_cm();
    if (cm==0){
        cm=500;
    }
    return cm;
}

void moveStop(){
```

```
digitalWrite(RightMotorForward, LOW);
digitalWrite(LeftMotorForward, LOW);
digitalWrite(RightMotorBackward, LOW);
digitalWrite(LeftMotorBackward, LOW);
}

void moveForward() {

    if(!goesForward){

        goesForward=true;

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorBackward, LOW);
    }
}

void moveBackward() {

    goesForward=false;

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorForward, LOW);

}

void turnRight() {

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorForward, LOW);

    delay(9000);
}
```

```
digitalWrite(RightMotorForward, LOW);
digitalWrite(LeftMotorForward, LOW);
digitalWrite(RightMotorBackward, LOW);
digitalWrite(LeftMotorBackward, LOW);
}

void moveForward() {

  if(!goesForward) {

    goesForward=true;

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
  }
}

void moveBackward() {

  goesForward=false;

  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorForward, LOW);

}

void turnRight() {

  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorForward, LOW);

  delay(9000);
}
```

```
digitalWrite(LeftMotorForward, HIGH);
digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);
digitalWrite(RightMotorBackward, LOW);

}

void turnRight2() {

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorForward, LOW);

    delay(18000);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);

}

void turnLeft() {

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);

    delay(9000);

    void turnLeft2() {

        digitalWrite(LeftMotorBackward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorForward, LOW);
        digitalWrite(RightMotorBackward, LOW);

        delay(18000);

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorBackward, LOW);

    }
}
```

Figura 56. Código para evitar colisiones y seguir la ruta